

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

«До захисту допущено»
Завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

“ _____ ” _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ
освітньо-кваліфікаційного рівня "Спеціаліст"
зі спеціальності 7.05010201 "Комп'ютерні системи та мережі"
на тему:

PHR-ОРІЄНТОВАНА СИСТЕМА ІНТЕРАКТИВНОГО КЕРУВАННЯ ВЕБ-РЕСУРСАМИ

Студент
групи КСМзскп-51 _____ Стасюк І.Г.
(підпис)

Керівник:
к.ф.-м.н., доцент _____ Касянчук М.М.
(підпис)

Нормоконтроль
к.т.н., доцент _____ Васильків Н.М.
(підпис)

Консультант
з охорони праці
доцент _____ Сапожник Г.В.
(підпис)

2012

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії
спеціальність 7.05010201 – “Комп'ютерні системи та мережі”

“Затверджую”
завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

“ ___ ” _____ 20__ р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТА**
Стасюка Ігоря Григоровича

- 1. Тема проекту:** "RНР–орієнтована система інтерактивного керування веб–ресурсами" затверджена наказом університету № 475 від 14 жовтня 2012 р.
- 2. Термін здачі студентом закінченого проекту** “ ___ ” _____ 20__ р.
- 3. Вихідні дані для проекту:** Технічне завдання.
- 4. Перелік задач, які мають бути вирішені:**
 - провести аналіз існуючих рішень задачі, особливостей структур та функціональності інтерактивного керування веб–ресурсами;
 - розглянути основні архітектурні та технічні помилки при розробці інтерактивного керування веб–ресурсами;
 - розробити семантичну модель програмних ресурсів;
 - розробити нормалізовану та концептуальну модель програмних ресурсів;
 - встановити необхідну умову коректності програмних ресурсів;
 - розробити структуру програмного забезпечення інтерактивного керування веб–ресурсами;
 - виконати тестування програмного сервісу.
- 5. Перелік графічного матеріалу (з точним вказанням обов'язкових креслень)**
 - Нормалізована модель програмних ресурсів. Схема структурна
 - Концептуальна модель програмних ресурсів
 - Схема взаємозв'язку програмних модулів
 - Додавання нової сторінки та її редагування. Схема структурна

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

6.Консультанти по проекту (із зазначенням розділів):

Розділ	Консультант	Підпис
Охорона праці	Сапожник Г.В.	

КАЛЕНДАРНИЙ ПЛАН

№	Назва розділів дипломного проекту	Термін виконання	Позначки керівника про виконання завдань
1	Особливості систем звукового забезпечення	15.09.2011 – 5.11.2011	
2	Проектування мережі звукового забезпечення	6.11.2011 – 31.01.2012	
3	Апаратно–програмна реалізація системи звукового забезпечення	1.02.2012 – 14.04.2012	
4	Охорона праці	15.04.2012 – 23.04.2012	

Завдання прийняв до виконання _____
(підпис)

Керівник дипломного проекту _____
(підпис)

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

АНОТАЦІЯ

Робота виконана на 91 сторінці, з них 84 сторінки основного тексту, містить 40 рисунків, 3 таблиці, 18 джерел посилань на 2 сторінках, 5 додатків.

Метою дипломного проекту є розробка РНР-орієнтованої системи інтерактивного керування веб-ресурсами.

У дипломному проекті проаналізовано принципи розробки систем інтерактивного керування веб-ресурсами і їх недоліки, на основі чого був розроблений програмний сервіс.

Досліджено структуру та функціональність систем. Проаналізовано подібні програмні комплекси, їх переваги і недоліки. Було проведено розробку моделі та схеми вимірювання якості програмних ресурсів. Розроблено структуру програмного сервісу. Здійснено опис основних алгоритмів і процедур програмного сервісу. Розроблений програмний сервіс є гнучким та універсальним, тому в подальшому можна розширювати його функціональність.

Дипломний проект містить структурну схему нормалізованої моделі програмних ресурсів; концептуальна модель програмних ресурсів; схема взаємозв'язку програмних модулів; структурну схему додавання нової сторінки та її редагування, які подані в графічній частині.

Дипломний проект має практичну спрямованість і його результати плануються до впровадження.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

ANNOTATION

The work performed at 91 pages, including 84 pages main text, contains 40 figures, 3 tables, 18 references sources on 2 pages, 5 applications.

The aim of the diploma project is to develop PHP-based system to manage online web resources.

In the thesis project analyzes the principles of development control interactive web resources and their shortcomings, based on what was designed software service. The structure and functionality of systems. Analysis of similar software systems, their advantages and disadvantages. There were development models and measurement schemes as software resources. The structure of software services. Done description of the basic algorithms and procedures for program services. Developed a software service is a flexible and versatile, so you can further expand its functionality.

The degree project contains block diagram normalized model program resources, conceptual model of program resources; circuit interconnection software modules; diagram adding a new page and edit it, which are presented in graphical part.

The diploma project is practically oriented and its results are planned for implementation.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Технічне завдання

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 РНР–орієнтована система інтерактивного керування веб–ресурсами

1.2 Область застосування – провайдери Internet, установи, які мають власні сайти, державні установи, редакції газет, в яких необхідна система інтерактивного керування веб–ресурсами.

2. ОСНОВА ДЛЯ РОЗРОБКИ

Основою для розробки є завдання на дипломний проект, затверджене кафедрою комп'ютерної інженерії факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою дипломного проекту є розробка РНР–орієнтованої системи інтерактивного керування веб–ресурсами

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами даної розробки є матеріали навчальної та реферативної наукової літератури, технічна документація, існуючі програмні та програмно-апаратні системи, журнали, науково-дослідні роботи вітчизняних та закордонних вчених.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до апаратних засобів

5.1.1 Функціональні вимоги до апаратних засобів.

5.1.1.1 Система повинна працювати на IBM-сумісних робочих станціях.

5.1.1.2 Мінімальні вимоги до робочих станцій: процесор від 1 ГГц, оперативна пам'ять від 512 Мб, відеокарта від 32 Мб, об'єм пам'яті на жорсткому диску до 100 Мб, клавіатура, маніпулятор «миша».

5.1.1.3 Відеореєструюча апаратури (відеокамера, фотокамера та їх роздільна здатність, сумісність програмного забезпечення).

5.2 Вимоги до програмної системи

5.2.1 Функціональні вимоги до програмної системи

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

5.2.1.1 Оператор системи повинен мати змогу виконувати наступні функції:
- інтеграція з іншими інженерними і комунікаційними системами об'єкта;
- здійснення управління веб-ресурсами через програмний комплекс;
- формування звітів на основі проведених досліджень у табличному та графічному форматах.

5.2.1.2 Вхідна інформація отримується шляхом:

– завантаження текстових та графічних файлів із цифрових носіїв даних: жорстких дисків, гнучких дисків, flash-карт тощо;

– отримання файлів за допомогою відповідної апаратури у реальному часі.

5.2.1.3 Вихідна інформація:

– вихідна інформація повинна подаватись у простому та інтуїтивно зрозумілому для користувача форматі;

– формування звітів повинно відбуватись у реальному часі;

– вихідна інформація виводиться у текстовому, табличному, графічному (графіки, діаграми) форматах;

5.2.2 Вимоги до надійності.

5.2.2.1 Передбачити контроль введеної інформації.

5.2.2.2 Розробити комплекс заходів контролю коректності дій користувача під час роботи з системою.

5.2.2.3 Забезпечити можливість відновлення роботи системи після збоїв.

5.2.3 Вимоги до програмного забезпечення:

5.2.3.1 Операційна система сімейства Windows;

5.2.3.2 Графічна бібліотека OpenGL.

5.2.3.4 Сумісність з сучасними форматами даних:

– вхідна інформація подається у форматах «*.bmp», «*.jpg», «*.docx»;

– вихідна інформація подається у форматах «*.txt» та «*.doc» для текстової інформації, «*.xls» для табличної інформації та «*.bmp», «*.jpg» для графічної.

5.2.4 Вимоги до програмної документації

5.2.4.1 Код програмних модулів повинен містити необхідні для його розуміння коментарі;

5.2.4.2 Розроблене програмне забезпечення повинно включати довідкову систему

5.2.5 Вимоги експлуатації

5.2.5.1 Кліматичні вимоги до експлуатації, при яких забезпечується робота програми повинні відповідати кліматичним умовам експлуатації наявних технічних засобів

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

5.2.5.2 Вимоги до кваліфікації та численності персоналу. Мінімальна кількість персоналу, необхідного для роботи програми, може складати одну штатну одиницю – кінцевого користувача програми – оператор.

5.2.6 Вимоги до захисту:

5.2.6.1 Мінімальна довжина пароля - 10 символів.

6. ВИМОГИ ОХОРОНИ ПРАЦІ

В розділі “Охорона праці ” дипломного проекту повинен бути даний аналіз умов праці в приміщенні де працює розробник апаратно–програмного засобу.

7. ПОРЯДОК КОНТРОЛЮ І ПРИЙОМКИ

7.1 Представлення дипломного проекту на попередній захист

7.2 Представлення дипломного проекту на захист

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

ЗМІСТ

Вступ.....	10
1 Структура та функціональність інтерактивного керування веб-ресурсами.....	12
1.1 Поняття та функції інтерактивного керування веб-ресурсами.....	12
1.2 Структура розробки інтерактивного управління веб-ресурсами.....	18
1.3 Структура даних.....	24
1.4 Розповсюджені архітектурні і технічні помилки при розробці інтерактивного керування веб-ресурсами.....	26
1.5 Аналіз існуючих аналогів.....	27
2 Семантична модель програмних ресурсів.....	32
2.1 Модель та схема вимірювання якості програмних ресурсів.....	32
2.2 Програмні ресурси як семантичне відображення.....	34
2.3 Нормалізована модель програмних ресурсів.....	35
2.4 Семантичний простір.....	36
2.5 Розширений семантичний простір.....	41
2.6 Необхідна умова коректності програмних ресурсів.....	43
2.7 Концептуальна модель програмних ресурсів.....	44
2.8 Семантична модель дефектів операндів.....	47
3 Розробка та опис програмного сервісу інтерактивного керування веб-ресурсами.....	51
3.1 Структура розробленого програмного забезпечення інтерактивного керування веб-ресурсами.....	51
3.2 Функціональність розробленого програмного забезпечення.....	52
3.3 Тестування програмного сервісу.....	62
3.4 Захист програмного сервісу.....	67

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			16

4 Охорона праці.....	71
Висновки.....	84
Список використаних джерел.....	85
Додаток А. Нормалізована модель програмних ресурсів. Схема структурна.....	87
Додаток Б. Концептуальна модель програмних ресурсів.....	88
Додаток В. Схема взаємозв'язку програмних модулів.....	89
Додаток Г. Додавання нової сторінки та її редагування. Схема структурна.	90
Додаток Д. Довідка про впровадження.....	91

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Коли Інтернет тільки почав своє існування, з'явилися перші веб-сайти [1]. Їх підтримка та постійне оновлення вимагали знань мови гіпертекстової розмітки (HTML) та навичок роботи з графічними зображеннями.

Однак ці веб-ресурси давно застаріли, вони склалися з декількох сторінок тексту і пари-трійки картинок. На сьогоднішній день практично кожен веб-сайт має, як мінімум, декілька різних розділів і підрозділів, безліч графіки, флеш-анімації, музики і відеофайлів на своїх сторінках. Управління всім цим розмаїттям часом і від професійного веб-майстра вимагає чималих знань і витрат часу, а що ж тоді говорити про власників сайту - людей, часто не володіють необхідним багажем цих самих знань та й часу теж не мають в своєму розпорядженні?

Очевидно, що необхідний якийсь інструмент, за допомогою якого завдання управління сайтом значно спрощується.

Системи управління веб-ресурсами, з одного боку, являє інтерфейс користувача - відвідувача сайту, де господар показує, що у нього є, організує різні WEB-сервіси, а відвідувач може дивитися, купувати, спілкуватися. З іншого боку - це інтерфейс адміністраторській панелі, недоступний для відвідувачів, де і відбувається управління ресурсом - адміністрування, управління виглядом сайту, спілкування з відвідувачами та клієнтами WEB-сайту.

Ролі систем управління веб-ресурсами відводиться значна частина в загальному розвитку Інтернету. Всесвітня мережа постійно розвивається семимильними кроками, чому сприяють і загальна комп'ютеризація, і зростаюча зв'язок offline-світу і бізнесу з online способами доставки інформації. Виникає все більша кількість бажаючих мати своє представництво в Інтернеті.

Але, так як і будь-яке програмне забезпечення, системи інтерактивного керування веб-ресурсами теж мають свої недоліки. Наприклад, потрібно, щоб на веб-ресурсі виконувалися додаткові сервіси, яких не має в системі [2].

									Арк.
									18
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

Тут і виявляється перший недолік, усунути який можна кількома шляхами. Можна внести зміни в код самої системи, але для початку необхідно розібратися в її структурі. А якщо людина не є програмістом, або якщо на це немає часу, то все одно доведеться залучати фахівця з боку, який навряд чи буде братися за цю роботу, так як ці дії можуть порушити роботу всього сайту. Якщо все-таки, цю роботу хтось і візьметься робити, то коштувати вона буде недешево. З цього приводу у програмістів є прислів'я: «краще написати заново, ніж виправляти чужий код» [3].

Другий недолік - це надмірність. Адже ці системи створюються для вирішення самих різних проблем, серед яких будуть не обов'язково потрібні вам. Непотрібний код буде сповільнювати роботу всієї системи і про швидкість роботи веб-ресурсу можна буде забути [4].

Третій недолік - це проблеми з оптимізацією веб-ресурсів [5]. При використанні систем управління веб-ресурсами його оптимізація іноді стає неможливою.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

1 СТРУКТУРА ТА ФУНКЦІОНАЛЬНІСТЬ ІНТЕРАКТИВНОГО КЕРУВАННЯ ВЕБ-РЕСУРСАМИ

1.1 Поняття та функції інтерактивного керування веб-ресурсами

1.1.1 Поняття інтерактивного керування веб-ресурсами

Будь-який Web-сайт складається з набору відповідних сторінок, а відмінності полягають лише в тому, як ці сторінки організовані. Існує два види організації Web-сайту - статичний і динамічний. У першому випадку фахівці, що відповідають за створення та підтримку сайту, повинні написати в HTML-форматі кожен окремо сторінку, включаючи її оформлення і контент. У другому випадку - в основі будь-якої Web-сторінки лежить заданий шаблон, який визначає розташування у вікні Web-браузера всіх компонентів сторінки і вставка конкретної інформації проводиться з використанням стандартних вбудованих засобів, що не вимагають від учасника процесу знання мови HTML і досить складних для людини, яка не є спеціалістом у процедурах публікації Web-сторінки.

Якщо сайт складається з безлічі сторінок або він повинен часто оновлюватися, то перевага динамічної організації стає очевидним. Розробникам Web-сайту не треба переписувати всю сторінку при зміні її інформаційного наповнення або дизайну. Сторінки не зберігаються цілком, а формуються динамічно при зверненні до них.

Таким чином, відділення дизайну від контенту є головною відмінною особливістю динамічних сайтів від статичних. На цій основі можливі подальші удосконалення структури сайту, такі як визначення різних призначених для користувача функцій і автоматизація бізнес-процесів, а саме головне, контроль що надходить на сайт контенту [6].

Для створення динамічного сайту можливі два шляхи. По-перше, це написання власних програм, які відповідають за створення потрібних шаблонів і підтримуючих необхідні функції. При цьому створена система буде повністю відповідати потребам, проте можливо вимагатиме великих програмістських

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

зусиль і часу. Другий шлях - це скористатися вже існуючими системами, які і називаються системами управління веб-ресурсами. Перевагою цього шляху є зменшення витрат часу і сил. До його недоліків можна віднести зниження гнучкості, надання недостатнього або надмірного набору можливостей. Але структура роботи системи управління контентом приблизно однакова [7] і представлена на рисунку 1.1.

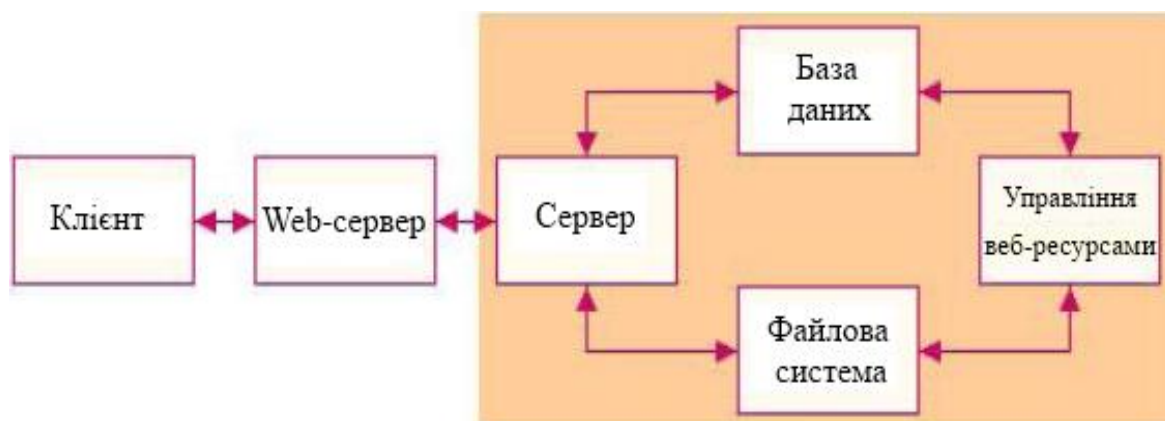


Рисунок 1.1 - Структура системи управління веб-ресурсами

Під контентом (дослівний переклад англійського терміна content, що означає зміст, вміст) розуміють інформаційне наповнення сайту - тобто всі типи матеріалів, які знаходяться на сервері: web-сторінки, документи, програми, аудіо-файли, фільми і так далі. Таким чином, управління контентом - це процес управління подібними матеріалами. Він включає наступні елементи: розміщення матеріалів на сервері, видалення матеріалів з сервера, коли в них більше немає необхідності, організацію (реорганізацію) матеріалів, можливість відслідковувати їх стан.

Системи управління контентом (в англійській мові існує усталений термін - Content Management Systems або, скорочено, CMS) - це програмні комплекси, що автоматизують процедуру управління контентом [8].

1.1.2 Функції інтерактивного керування веб-ресурсами

Функції інтерактивного керування веб-ресурсами можна розділити на кілька основних категорій [9]:

1) створення - надання авторам зручних і звичних засобів створення контенту;

2) управління - зберігання контенту в єдиному сховищі. Це дозволяє стежити за версіями документів, контролювати, хто і коли їх змінював, переконуватися, що кожен користувач може змінити тільки той розділ, за який він відповідає. Крім того, забезпечується інтеграція з існуючими інформаційними джерелами та ІТ-системами. Керування веб-ресурсами підтримує контроль над робочим потоком документів, тобто контроль за процесом їх схвалення. Таким чином, управління контентом включає в себе зберігання, відстеження версій, контроль за доступом, інтеграцію з іншими інформаційними системами і керування потоком документів.

3) публікація - автоматичне розміщення контенту на терміналі користувача. Відповідні інструменти автоматично адаптують зовнішній вигляд сторінки до дизайну всього сайту.

4) представлення - додаткові функції, що дозволяють поліпшити форму представлення даних; наприклад, можна будувати навігацію по структурі репозиторія.

Системи управління веб-ресурсами діляться на чотири основні категорії, які частково перекриваються:

1) системи управління вихідними кодами традиційно підтримують управління вихідними кодами програм, і часто надають деякий web-інтерфейс, який може використовуватися усередині корпоративної мережі, а також поза її для паралельної роботи з вихідними кодами;

2) системи управління документами призначені для організацій, що оперують з великою кількістю документів, наприклад, офіси великих компаній, редакції і страхові компанії;

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			22

3) системи управління web-контентом представляють собою нову індустрію програмних продуктів. Ці системи призначені для розробки й управління Web-сайтами різного ступеня складності. Зазвичай такі системи підтримують і деякий тип управління потоками робіт;

4) системи електронної комерції - забезпечують зберігання і керування електронними каталогами товарів. По суті, ці системи не дуже відрізняються один від одного. Найголовніша відмінність цих систем - це люди, які їх використовують.

Використання інтерактивного керування веб-ресурсами надає наступні переваги [10]:

1) оперативне оновлення інформації - інформацію публікує співробітник, що володіє інформацією, без додаткових посередників у вигляді технічних фахівців. Інтерактивне керування веб-ресурсами призначене для автоматизації процес публікації інформації на web-сайті, надаючи користувачам можливість самим публікувати матеріали в WWW і визначати їх візуальне подання, використовуючи для цього стандартні засоби, що не вимагають знання мови HTML і досить складних для неспеціаліста процедур. За допомогою керування веб-ресурсами можна, не будучи професійним розробником, створювати і модифікувати інформаційне наповнення сайтів;

2) зниження вартості підтримки - оновлення інформації проводиться самостійно, немає необхідності оплачувати працю власного або зовнішнього web-майстра. Зниження вартості відбувається за рахунок зниження витрат часу на пошуки документів, припинення дублювання і помилок, збільшення швидкості зв'язку з партнерами і клієнтами;

3) надання додаткових сервісів користувачеві - частина сервісів - пошук, форуми, голосування і т.д., вимагають інтерактивної взаємодії з користувачем;

4) зменшення термінів і вартості розробки - найбільш затребувана функціональність вже реалізована в системі керування веб-ресурсами і може бути використана відразу;

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

5) підвищення якості розробки - при розробці повністю або частково використовуються готові модулі, які вже пройшли неодноразове тестування;

б) зниження вартості подальших модифікацій – інтерактивне керування веб-ресурсами дозволяє розділити дані та їх подання. Це дозволяє набагато простіше змінити зовнішній вигляд сайту, ніж у випадку зі статичним сайтом.

Розробкою систем управління контентом займаються багато компаній, у тому числі IBM, Microsoft, Oracle, Macromedia [11].

1.1.3 Модель представлення даних

Існує класифікація програмного забезпечення інтерактивного керування веб-ресурсами, заснована на моделі представлення даних - об'єктної, мережевий або модульної.

Об'єктна модель представлення даних оперує такими поняттями, як клас і об'єкт. Класи визначають структуру даних і являють собою набір атрибутів (текстовий рядок, ціле число, зображення і т.д.). Екземпляри класу (об'єкти) мають певну структуру і можуть містити інші об'єкти, утворюючи довільну ієрархічну структуру. Об'єкти можуть наслідувати властивості, зміст і поведінку об'єктів, які в них містяться. Прикладами об'єктів служать документи, картинки, папки і облікові записи користувачів. Зв'язки між об'єктами створюються, наприклад, за допомогою таблиць вигляду `id`, `from_object`, `from_object_version`, `to_object`. На рисунку 1.2 приведена можлива структура класу «Стаття» та його представника.

У керування веб-ресурсами дані зазвичай зберігаються в реляційній або об'єктної базі даних. У першому випадку об'єктна модель даних відображається на реляційну модель бази даних.

Як правило, системи, що ґрунтуються на об'єктно-орієнтованій моделі даних, найбільш функціональні, гнучкі, але, в той же час, і характеризуються найбільшою складністю.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

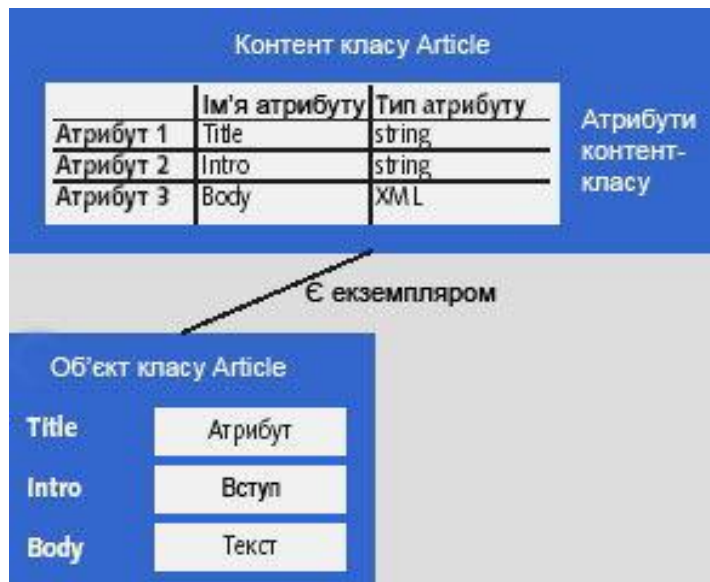


Рисунок 1.2 - Приклад організації контенту за допомогою об'єктної моделі даних

Мережева модель представлення даних (рисунок 1.3) спирається на теорію графів: структура інформації представляється у вигляді вузлів з поміченими зв'язками між ними. Фундаментом системи може служити як мережева, так і традиційна реляційна СУБД, на яку відображена мережева модель опису даних. У реляційних таблицях зберігається інформація про вузли, їх атрибути і зв'язках між ними. Зв'язок відрізняється від атрибуту тим, що в ній зберігається посилання на інший вузол, а в атрибуті - власне значення. Для отримання даних з спрямованого графа зазвичай використовуються рекурсивні процедури обробки, такі як складання списків вузлів, визначення атрибутів вузла по атрибутам батька та ін.

У модульній моделі контент розділений на окремі модулі за типами вмісту. Структура даних залежить від модуля, і вся робота з контентом зосереджена всередині модуля. Модулі незалежні і повністю відповідають за роботу з документами даного типу. Документи описуються з допомогою фіксованого набору характеристик - типи документів суворо фіксовані. Розширювати функціональність можна за рахунок додавання нового модуля, заміни або редагування існуючого коду. Найчастіше немає ніякої системи

зв'язків між документами різних модулів і між документами одного і того ж модуля. Стандартний набір типів контенту (модулів) такий: посилання, статті, файли, новини, розділи, форум.

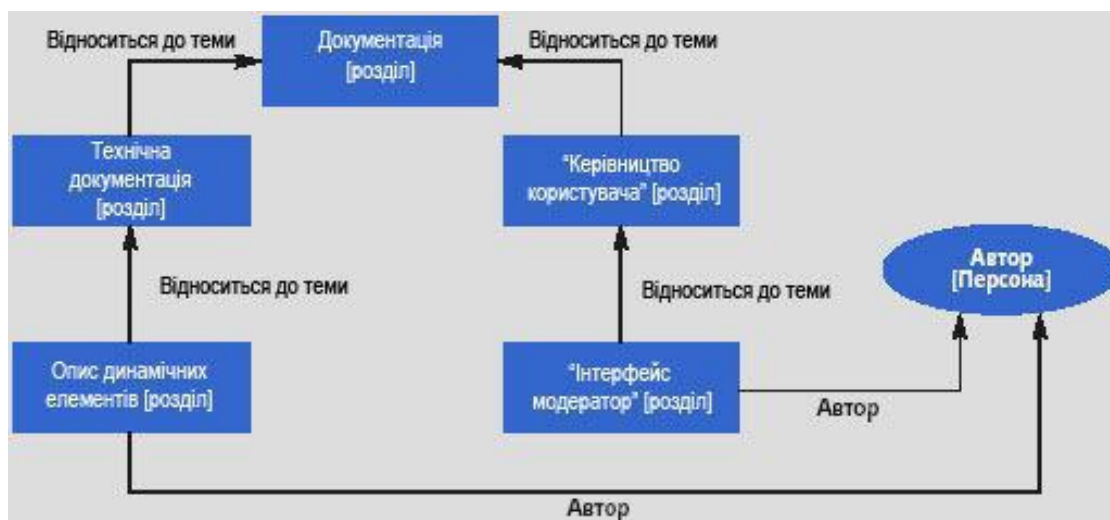


Рисунок 1.3 - Приклад організації контенту за допомогою мережевої моделі даних

Незважаючи на очевидну обмеженість моделі даних, системи на її основі найбільш популярні завдяки своїй простоті. У модульних систем керування веб-ресурсами є один спільний недолік - строго фіксована в межах модуля структура вмісту. Проте для розширення їх функціональності можна скористатися відповідними зовнішніми модулями. Очевидна перевага цих систем - можливість одержання за короткий час готового до використання порталу.

1.2 Структура розробки інтерактивного управління веб-ресурсами

При розробці веб-ресурсів для більш ефективної роботи потрібно слід точно визначити, що потрібно розробити, з яких файлів повинен складатися сайт [12].

А для розробки системи інтерактивного керування веб-ресурсами потрібно розробити її структуру і принцип роботи. Функціональні можливості

системи управління веб-ресурсами є визначальним фактором при виборі нового продукту. Не меншу значимість має зручність його використання.

Якщо персонал, особливо автори контенту, не мають можливості без праці використовувати систему управління веб-ресурсами, така система ніколи не буде успішною, незважаючи на спектр її можливостей.

Схема роботи системи інтерактивного керування веб-ресурсами працюють на основі зв'язків, представлених на рисунку 1.4.

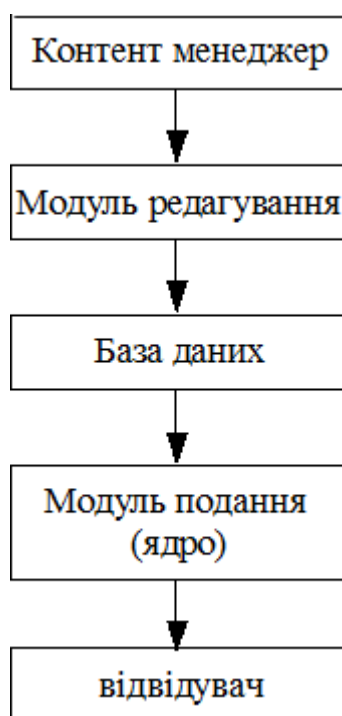


Рисунок 1.4 - Схема роботи системи інтерактивного керування веб-ресурсами, зображена на основі зв'язків.

Інформація в базі даних змінюється за допомогою модуля редагування [13].

Модуль подання генерує сторінку із змістом при запиті на нього на основі інформації з бази даних. Сторінки наново створюються сервером (модулем подання) при кожному запиті. На основі URL у модулі запиту відбувається визначення запитуваної порції контенту/сторінки (наприклад через параметри методу GET).

Загальна зручність використання, а також те, наскільки система відповідає вимогам користувачів, є факторами, що здобувають усе велике значення в процесі оцінки і вибору продукту.

Зручна система управління веб-ресурсами:

- має мінімальний набір елементів;
- працює стабільно і безпомилково;
- має інтерфейс, заснований на описі задач;
- не навантажує користувача технічними подробицями;
- відповідає основним принципам юзабіліті;
- відповідає моделі мислення авторів текстів;
- зручна як для рідкісного, так і щоденного використання;
- має ефективний користувальницький інтерфейс;
- дає доступ до інструкцій;
- не вимагає тривалого навчання використанню;
- самостійне використання.

Розберемо найголовніші критерії системи управління веб-ресурсами.

1.2.1 Мінімальний набір елементів

З нарощуванням функціональності зростає кількість кнопок, пунктів меню та посилань.

Авторів (як і сторонніх адміністраторів) може легко збити з пантелику такий вибір, що збільшує витрати часу на освоєння системи і створює серйозні проблеми зі зручністю використання. Тому важливий такий принцип: необхідно зменшувати число видимих елементів інтерфейсу.

Частково це завдання можна вирішити за допомогою поділу ролей та обмеження прав при використанні системи звичайними авторами, що дасть моментальне спрощення авторських інтерфейсів.

Проте крім цього, система управління веб-ресурсами повинна бути влаштована таким чином, щоб спростити інтерфейс можна було в будь-який момент.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Цей пункт включає можливості:

- прибрати опції, не пов'язані з окремим користувачем або невикористовуванні ними;
- обмежити використання меню і випадаючих списків, які містять загальні опції;
- встановити доступ до системи управління веб-ресурсами, заснований на розмежуванні ролей користувачів;
- використовувати принцип опису завдань у створенні внутрішніх інтерфейсів.
- обмежити кількість функцій системи, що надаються авторам.

1.2.2 Стабільна і безпомилкова робота

Думаємо, не варто навіть згадувати, що автори не повинні втрачати наполовину написані матеріали з-за проблем з системи управління веб-ресурсами.

У реальному житті домогтися цього на 100% нелегко. Веб-додатки за своєю суттю менш стійкі до впливу користувачів або системним проблемам, тому для подолання даної труднощі належить виконати ще багато роботи.

Крім цього в системі управління веб-ресурсами повинні бути присутнім наступні можливості:

- автозбереження контенту за будь-якої можливості
- забезпечення збереження написаного при збої в браузері або CMS
- стійкість при виникненні помилок
- відображення ясних (і легким для читання) повідомлень про

помилки

Тобто автори та власники сайтів не повинні завжди бути наготові, щоб уникнути потенційних проблем при роботі з CMS. Навпаки, це CMS повинна створювати якомога менше проблем, і гідно справлятися з ними, навіть якщо вони з'являються. Помилки та збої не повинні вести до втрати написаного тексту.

									Арк.
									29
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

1.2.3 Інтерфейс, що ґрунтується на описі задач

У загальному розумінні призначення систем управління веб-ресурсами зводиться до виконання певних завдань, будь то створення нової сторінки або реєстрація нового автора.

Можна поліпшити роботу системи управління веб-ресурсами, використовуючи інтерфейс, заснований на описі типових завдань, що вирішуються цією системою.

Часто маючи форму «майстра», такий інтерфейс може спростити роботу, а також значно скоротити набір опцій, доступних в кожен момент роботи з системою.

Як мінімум, весь набір функцій повинен бути розділений на декілька великих категорій, відповідних способів використання системи. Наприклад, внутрішні адміністраторські завдання (безпека або властивості користувачів) повинні розташовуватися окремо від часто використовуваних завдань з написання контенту.

На практиці ж виробники найчастіше спокушаються «елегантністю» загальної структури системи, що найчастіше означає наявність тільки одного способу виконання всіх завдань, побудованого навколо основного набору об'єктів і меню.

Незважаючи на елегантність такого підходу, часто його використання призводить до того, що інтерфейс віддаляється від принципу опису завдань і стає складним при виконанні авторами звичайних дій. Для демонстрації клієнтам, наскільки добре дотриманий у вашій системі принцип опису задач, слід використовувати відповідні сценарії. Авторам необов'язково бачити «закулісся» системи.

1.2.4 Ефективний користувацький інтерфейс

Крім легкості у використанні, система управління веб-ресурсами повинна володіти ефективністю, особливо при регулярній роботі з нею.

На практиці ефективність означає:

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

- зменшення кількості кліків мишею для виконання завдання;
- зменшення кількості сторінок і вкладок, за допомогою об'єднання елементів;
- ефективне використання елементів вікон інтерфейсу;
- зменшення кількості необхідних оновлень сторінки;
- створення коротких і ефективних шляхів виконання частих операцій;
- збільшення швидкості роботи і швидкості реагування CMS (актуально із збільшенням використання)

Класичним прикладом низької ефективності в багатьох системах управління веб-ресурсами можна вважати розростання кількості вкладок. Одна вкладка використовується для стандартних операцій з сторінкою (наприклад, публікації дати), інша - для роботи з контентом, а третя - для додаткових дій. У цій ситуації користувачу потрібно часто перемикатися між вкладками і вводити кілька разів одну й ту ж інформацію. При кожній зміні вкладки, система потребує оновлення сторінки, що тягне додаткове уповільнення роботи.

Звичайно, це не заклик до щільного розміщення всіх опцій в одному місці, але варто приділити увагу балансу між простотою і ефективністю використання системи управління веб-ресурсами [5].

1.2.5 Мінімальний набір навичок і простота користування

На практиці в більшості своїй авторами контенту є звичайні офісні службовці. Тому система управління веб-ресурсами повинна бути налаштована таким чином, щоб мінімізувати час початкової підготовки авторів. Те ж стосується власників сайтів та адміністраторів, з урахуванням того, що для цієї категорії потрібна розширена підготовка.

Незважаючи на те, що зовсім позбутися від необхідності підготовки неможливо, система управління веб-ресурсами може бути влаштована так, щоб витрати на цю підготовку були абсолютно мінімальними.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

На практиці це означає створення інтерфейсів, які дозволяють в один клік виконувати такі прості операції, як:

- публікація нового контенту
- управління і реструктурування сайту
- додавання користувачів і керування ними
- створення або оновлення ділових процедур
- оновлення налаштувань безпеки

У всіх цих випадках користувач повинен обходитися без спеціальної технічної підготовки [14].

1.3 Структура даних

Перш, ніж говорити про структуру даних, потрібно визначити, яка структура повинна бути у сайту. Найбільш універсальна структура - звичайно ж, деревоподібна. У цьому випадку кожен розділ сайту, що входить в головне меню, може містити власну вкладену структуру внутрішніх розділів. Назвемо їх підрозділами або розділами другого рівня вкладеності. Головна сторінка, так само як і інші, що входять в головне меню, розглядається як сторінка верхнього (першого) рівня.

Заради простоти побудуємо алгоритм для «дерева», що містить всього 2 рівні, представлений на рисунку 1.5.

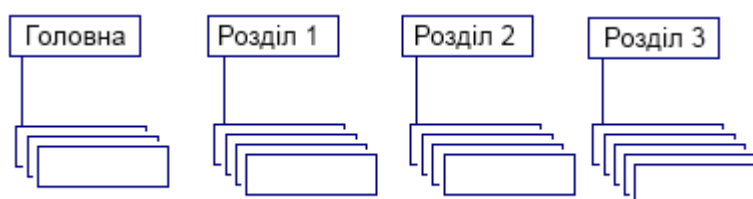


Рисунок 1.5 - Деревоподібна структура.

Але структуру даних нам ніхто не заважає побудувати так, щоб вкладеність розділів могла бути необмеженою, як у великих каталогів, де таке

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

«розгалуження» рубрик може досягати чималої глибини. Структура даних дозволяє створювати і зберігати скрипти, що робить легкою інтеграцію практично будь-яких готових рішень..

Почнемо з того, які дані нам потрібні для формування сторінки. Зверху вниз:

- рядок для тега <title>;
- рядок для мета-тега <keywords>;
- рядок для мета-тега <description>;
- HTML-код, що становить меню;
- рядок для заголовка на сторінці <h1>;
- текст для сторінки, включаючи HTML-теги розмітки та форматування;
- елементи оформлення сторінки, які складають шаблон. Як правило, для всіх сторінок сайту однаковий - порожня сторінка з дизайном, але без контенту. Набір даних для кожної сторінки можна зберігати в окремому файлі - в цьому випадку для формування сторінки достатньо мати тільки один файл, розділити дані і вставити їх в потрібні місця шаблону.

З даними сторінки все відносно просто. Беремо наступні рядки з нашого списку:

- рядок <title>;
- рядок <keywords>;
- рядок <description>;
- рядок заголовка <h1>;
- текст.

Збираємо все це в один рядок, поділяємо символом вертикальної риси | і записуємо у файл. Цей символ-роздільник обраний просто тому, що він практично ніколи не зустрічається в текстах. І все ж, щоб движок не помилився, якщо він все-таки зустрінеться в тексті, потрібно перед записом замінити всі знайдені в тексті роздільники на HTML-еквівалент, використовуючи код символу |. Замість переказів рядків вставляємо \ n - еквівалент, зрозумілий PHP.

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			33

При зчитуванні даних з файлу наш скрипт зобов'язаний буде відновити перекази рядків. Вона виглядає так:

це title | це ключові слова | це description | це заголовок h1 | А це витягнутий в один рядок текст ...

Що стосується меню, то цей елемент для кожної сторінки буде формуватися окремо - можливо. По-перше, в меню повинні бути посилання на сторінки, а список сторінок нам заздалегідь не відомий і буде змінюватися. По-друге, у нас буде не одне меню. У разі, якщо це сторінка розділу, що містить підрозділи, потрібно ще й меню нижнього рівня - список посилань на підрозділи [15].

1.4 Розповсюджені архітектурні і технічні помилки при розробці інтерактивного керування веб-ресурсами

Розмежування на рівні сутності БД елементів контенту і навігації (меню). Меню не є незалежною сутністю і фактично являє собою ієрархічний перелік посилання на елементи контенту, тобто вся інформація про меню міститься в об'єктах контенту.

Вихід: зберігати ієрархію сторінок в таблиці веб-сторінок за допомогою «рекурсивного зовнішнього ключа на цю ж таблиці» (дерево сторінок).

Використання в URL сторінки автоінкрементного ідентифікатора сторінки неефективне з точки зору usability. Допускається Лише для сайтів ЗМІ і подібних.

У якості унікального поля використовується текстовий ідентифікатор сторінки (наприклад: about, contacts, pronas, tovary).

Хоча властивість залежності URL сторінки від місця сторінки в структурі сайту зазначається як позитивна з точки зору usability (видно по URL, де ця сторінка в структурі), це має негативне значення для потреби гнучкості розвитку сайту і SEO (оптимізації під пошукові системи) - у разі зміни URL сторінка

									Арк.
									34
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

втрапить придбані раніше позиції в пошуковій системі, якщо URL сторінки не залежить від місця сторінки в ієрархії сайту.

У якості унікального ідентифікатора сторінки використовувати текстовий ідентифікатор (наприклад: about, contacts, pronas, tovary). У свою чергу, це позитивно відобразиться на usability.

Використання у якості URL сторінки транслітерованого заголовка сторінки негативно позначається для SEO: при зміні заголовка змінюється URL.

Багатомовність реалізується за допомогою додаткової сутності (таблиць) в БД. Це ускладнює як роботу програміста, так і роботу контент-менеджера, оскільки зберігати іншомовні версії потрібно в тій же таблиці сторінок у відповідних полях, що дублюють основні текстові поля для контенту [12].

1.5 Аналіз існуючих аналогів

1.5.1 Безкоштовний аналог 4site

Гнучкий, відкритий програмний продукт для розробки, підтримки і успішного розвитку інтернет-проекту та управління наповненням (контент-менеджмент). Всі сторінки сайту мають шаблони оформлення. Таким чином, розділяється контент (вміст) сторінок від їх оформлення. Зміна дизайну (редизайн) сайту стає простим завданням. Головна сторінка системи управління представлена на рисунку 1.6.

Завдяки широкому набору розроблених функціональних модулів можна в короткі терміни використовувати на сайті динамічні елементи. Новинні стрічки, галереї зображень, інформаційні блоки, каталоги та інші структури, зміст яких вноситься через зручний веб-інтерфейс.

Проведення рекламних кампаній. Можливе використання практично будь-яких рекламних носіїв: всі види банерів, Rich Media, текстові блоки і т. п. Збір статистики показів, кліків, обмеження по показах і зверненнях. Централізоване управління роботою рекламних інструментів.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35



Рисунок 1.6 - Головна сторінка системи управління

Створення онлайн-спільнот додає сайту інтерактивності. Можливі проведення опитувань на сайті, форми зворотного зв'язку, поштові розсилки. Використовує можливості кращих форумів за вибором користувача, створює онлайн-спільноти. Керує профілями користувачів, які реєструються на сайті для отримання персоналізованих сервісів.

Система управління сайтами 4Site дає практично необмежені можливості по створенню індивідуального дизайну сайту, зберігаючи при цьому переваги системи управління веб-ресурсами.

Система 4Site не зберігає весь контент (вміст) сайту в базі даних. Зазвичай сторінки зберігаються в звичайних html-файлах. З боку користувача та пошукових роботів сайт є статичним, тобто має адреси з домену та шляхи по сайту до файлу сторінки (<http://example.com/html/path/my.shtml>).

Шаблон сторінок включений в ці файли. Посилання на динамічні елементи (Меню, стрічка новин, галерея) реалізовані через стандартні засоби Apache - SSI-includes. При зміні шаблону або контенту сторінки з бази даних береться шаблон, а з форми - контент.

Деякі сторінки складаються тільки з шаблону та посилання на скрипт (наприклад, каталог, новини). При цьому у початковій сторінці зберігається звичайна адреса «статичної» сторінки. При введенні додаткових параметрів до адреси додаються CGI-приставки (<http://example.com/html/path/my.shtml?id=33>).
Всі інші елементи сторінок (картинки, flash-ролики, css) знаходяться у файловій системі сервера.

Як було відмічено вище, 4Site надає широкі можливості за варіантами взаємного розміщення сайту і системи управління. Залежно від вибору варіанта розміщення, змінюються вимоги до обладнання, програмного забезпечення, а також змінюється стиль управління сайтом. Основних варіантів чотири:

- 1) сайт і система управління в Інтернеті на одному фізичному сервері;
- 2) сайт і система управління в локальній мережі компанії і сайт в Інтернеті;
- 3) сайт і система управління в локальній мережі компанії і в Інтернеті;
- 4) сайт і система управління в Інтернеті на кількох фізичних серверах.

Сайт і система управління в Інтернеті на одному фізичному сервері є найпростішим і невимогливим варіантом. Для роботи такої зв'язки на хостингу бажана можливість запуску декількох підсайтів з окремими директоріями скриптів і документів. Не потрібно робота через SOAP. Технологія розробки сайту може бути тільки одна: всі зміни відбуваються в Інтернеті, відповідно відображаються на робочому сайті.

1.5.2 Комерційний аналог

Система управління веб-ресурсами WebDirector - комерційна система управління сайтом, орієнтована на завдання, що стоять перед корпоративним веб-сайтом. Її головна сторінка зображена на рисунку 1.7.

Система управління веб-ресурсами WebDirector забезпечує всі етапи роботи над сайтом:

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

- зручне створення корпоративних сайтів, порталів та інтернет-магазинів, що містять: інформаційні сторінки, форуми, каталоги товарів з кошиком покупок, банерні підсистеми, новинні стрічки, розсилки і т.д.;
- зручне управління сайтом силами співробітників компанії;
- ефективне залучення нових клієнтів компанії з Інтернет за допомогою високого відвідування створених сайтів на пошукових системах Yandex, Rambler, Google тощо.



Рисунок 1.7 - Головна сторінка системи управління веб-ресурсами WebDirector

Основні особливості програмного продукту WebDirector:

- наявність візуального редактора шаблонів, що дозволяє налаштувати зовнішній вигляд сайту (верстка та налаштування стилів) в режимі WYSIWYG;
- наявність SEO-модулів - всіх необхідних для просування сайту на пошукових системах інструментів, вбудованих в саму CMS.

Система керування веб-ресурсами WebDirector займає особливе місце серед інших систем управління сайтами та програм для створення сайтів.

										Арк.
										38
Змн.	Арк.	№ докум.	Підпис	Дата						

ДП.КСМ.19110/11.00.00.000.ПЗ

Система керування веб-ресурсами WebDirector - це більше, ніж просто система управління сайтом, це система створення, управління та просування сайту:

Подібно візуальним програмам для створення сайту (таким, як Dreamweaver), WebDirector дозволяє дизайнеру-графіку візуально працювати з дизайном майбутнього сайту: професійно підбирати колірну палітру, міняти розміри і склад зон сторінкового шаблону, їх наповнення, задавати кольору елементів, вставляти картинки, визначати користувальницькі стилі. Робота у візуальному редакторі шаблонів схожа на те, як працюють дизайнери в графічних і поліграфічних пакетах. Однак, на відміну від програм для створення сайту, все це робиться через веб-інтерфейс (хоча можна працювати і локально).

Подібно конструкторам сайтів, система керування вмістом WebDirector спрощує роботу із створення сайту, дозволяючи самим початківцям почати роботу над Інтернет-проектom з візуальною інструкцією готового шаблону, в той же час, не віднімаючи у досвідчених верстальників можливості працювати безпосередньо з кодом. На відміну від конструкторів сайтів, WebDirector не накладає жорстких обмежень на структуру cms шаблону і зовнішній вигляд шаблону. Змінювати можна все. Немає і жорсткої прив'язки до хостингу, оскільки система керування вмістом WebDirector не використовує бази даних. Перенесення системи управління веб-ресурсами на інший хостінг або на локальний комп'ютер здійснюється простим копіюванням файлів по FTP.

Подібно іншим систем, WebDirector, дозволяє зручно керувати вмістом сайту, включаючи інформаційні сторінки, новинні стрічки, форми, каталоги, форуми, пошук, інтернет-магазин та інший інтерактив. Система керування вмістом WebDirector дозволяє розвивати Інтернет-проект, підключаючи на сторінках чужі програмні модулі або саморобні скрипти.

За своїм функціоналом WebDirector - це чітко орієнтована на свою нішу система для створення і просування корпоративних веб-сайтів (до 1000 сторінок/елементів каталогу). Неefективно використовувати WebDirector для створення великих порталів або Інтернет-магазинів на кілька тисяч товарів.

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			39

2 СЕМАНТИЧНА МОДЕЛЬ ПРОГРАМНИХ РЕСУРСІВ

2.1 Модель та схема вимірювання якості програмних ресурсів

Сферою застосування незалежної верифікації програмних ресурсів (ПР) мають бути експертиза, сертифікація та ліцензування ПР критичного застосування у державних установах регулювання безпеки та якості товарів та послуг, що дозволить:

- підвищити якість експертизи відповідності ПР регулюючим вимогам на підставі диверсифікації технологій верифікації за показниками „достовірність”, „повнота”, „трудомісткість”;

- зменшити ризики аварійних ситуацій, пов’язаних із залишковими дефектами ПР, що є ключовим фактором забезпечення безпеки АЕС в довгострокових програмах розвитку атомної енергетики.

Повна модель якості ПР (згідно зі стандартами ISO/IEC 9126, ISO/IEC 14598) подана трьома моделями, що визначають на рисунку 2.1:

- внутрішню якість;
- зовнішню якість;
- якість у використанні.

Така диверсифікація моделей якості призначена для оцінювання якості ПР у різних середовищах реалізації на базових етапах, таких як:

- розробка специфікацій ПР;
- інтеграція ПР;
- реалізація на реальній платформі.

Необхідною умовою забезпечення якості є можливість за допомогою відповідних методик вимірювання атрибутів – фізичних або абстрактних властивостей.

Метрика, за визначенням, це сукупність методу та шкали вимірювання атрибуту або характеристики ПР. Загальна схема вимірювання якості ПР передбачає формування ієрархії значень «метрика (міра) – атрибут – підхарактеристика – характеристика» для всіх складових частин (ієрархії)

									Арк.
									40
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

архітектури та на їх підставі інтегральної оцінки якості конкретного програмного проекту.

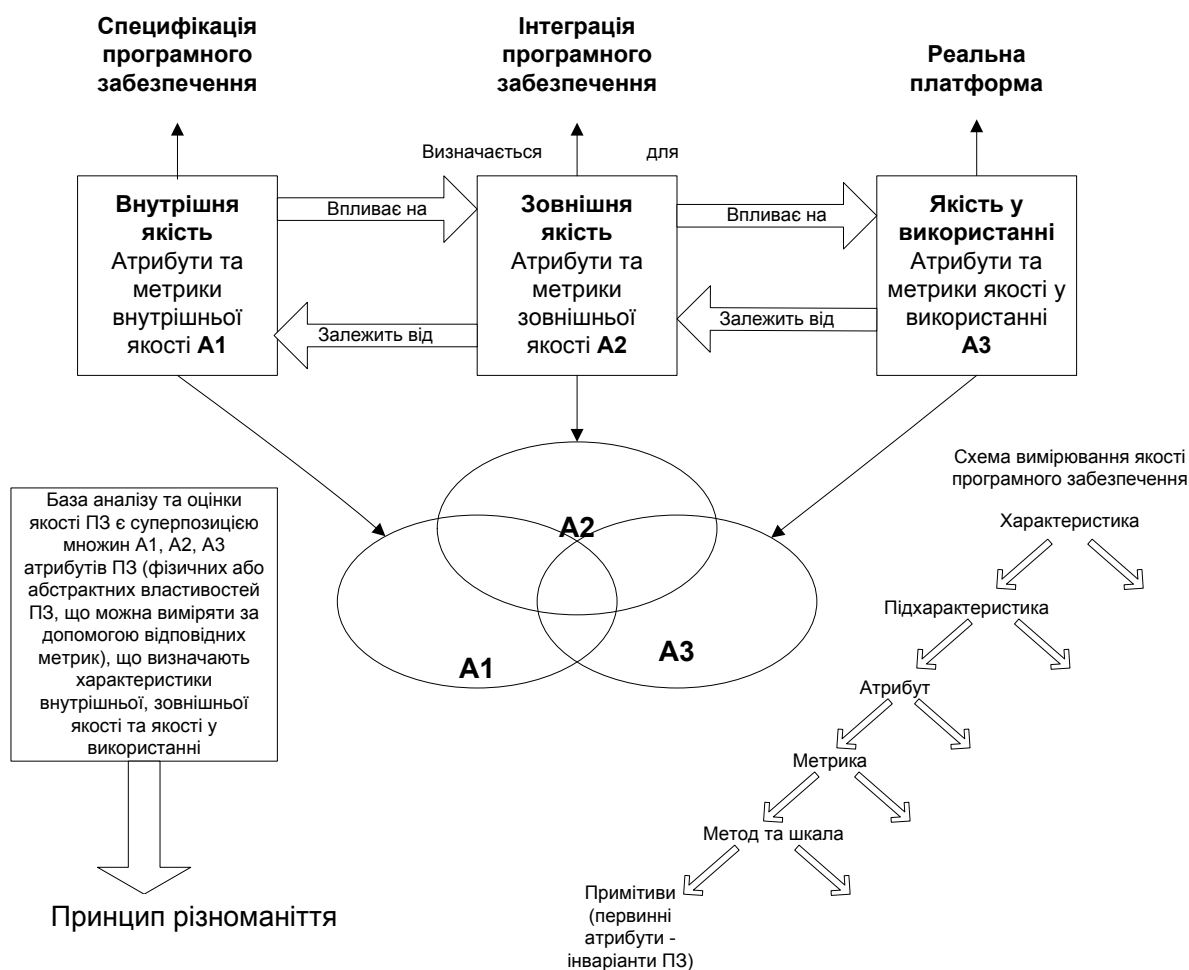


Рисунок 2.1 - Модель та схема вимірювання якості ПР

База аналізу та оцінки якості проекту ПР являють собою суперпозицію множин атрибутів ПР $A_1 \cup A_2 \cup A_3$, що визначають характеристики внутрішньої, зовнішньої якості та якості у використанні.

Пропонується доповнити модель вимірювання якості ПР контролем фізичних розмірностей програмних змінних, що відповідають вхідним даним та вихідним результатам.

Це дозволить підвищити надійність програмного забезпечення, тому що введений атрибут – фізична розмірність є інваріантом, а його порушення на будь-якому проектному рівні буде свідчити про порушення атрибутів якості.

2.2 Програмні ресурси як семантичне відображення

Диверсифікація технології оцінювання якості програмного забезпечення ґрунтується на розгляді відображень семантичних характеристик вхідних змінних області визначення у відповідні характеристики вихідних змінних області значень інформаційно-управляючих систем. Модель програмного забезпечення як семантичного відображення подана на рисунку 2.2.

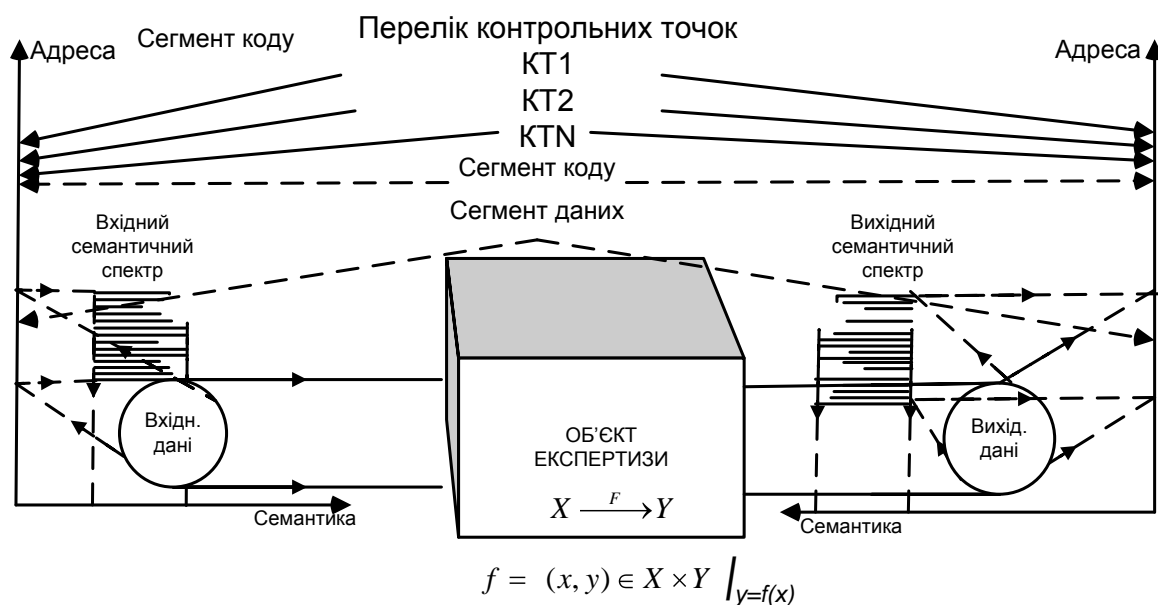


Рисунок 2.2 - Модель ПР як семантичного відображення

Області визначення і значень семантичного відображення, що реалізує ПР, представлені простором, визначеним у відповідному базисі.

Адресний простір ПР, розділений на сегменти коду і даних. Сегмент коду – програмні команди. Деякі з команд можуть бути контрольними, в яких на підставі спеціально розробленої алгебри здійснюватиметься контроль збереження семантичних інваріантів. Сегмент даних – програмні змінні. Кожна із змінних має деяку семантичну характеристику (фізичну розмірність), що є інваріантом, тобто не змінюється протягом всього часу роботи ІУС та при будь-яких вхідних даних.

Семантика – фізична розмірність програмної змінної (семантичний

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			42

інваріант). Вісь утворена проектуванням багатовимірного семантичного простору на пряму. Різним точкам осі „семантика” відповідають різні фізичні розмірності.

Семантичні спектри – лінії, що огинають відрізки, відповідають множині значень розмірностей вхідних даних та вихідних результатів у адресному просторі ПР. Спектр має проєкції на осі „Адреса” і „Семантика”. Відрізки різної довжини відповідають різним розмірностям (семантичним інваріантам). Позиція ліній по вертикалі відповідає адресі програмної змінної. На рисунку 2.2 показано два спектри: спектр вхідних даних та спектр вихідних результатів. Семантичне відображення зводиться до відображення вхідного семантичного спектру на вихідний семантичний спектр.

Для контролю семантичних інваріантів кожній скалярній програмній змінній та кожному елементу масиву ставиться у відповідність масив чисел (семантичний вектор), елементи якого відповідають проєкціям фізичної розмірності програмної змінної на базис семантичного простору. Це надає можливість спростити оцінку якості ПР та здійснювати її за допомогою незалежної верифікації шляхом статичного аналізу коду [11].

2.3 Нормалізована модель програмних ресурсів

У процесі незалежної верифікації програмний проект приводиться до нормалізованої форми – ієрархії програмно-технічних комплексів (ПТК), функцій і операторів, що проєктуються на адресний простір фон Неймана, як зображено в Додатку А.

Об’єкт експертизи складається з кількох ПТК, взаємодія яких умовно показана на функціональній моделі. Кожний з ПТК, відповідно до технічного завдання, виконує набір деяких функцій, взаємодія яких описується функціональними моделями. Інтерфейси всіх ПТК містять вхідні дані і вихідні результати. Кожна з програмно-реалізованих функцій ПТК є впорядкованою множиною операторів програмного коду, що функціонують у модифікованому

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			43

просторі фон Неймана, побудованою у базисі: адреса елементу пам'яті та умовний час. Весь обсяг адрес поділяється на два сегменти:

1) сегмент даних, що містить множину програмних змінних і констант, які мають семантичні характеристики;

2) сегмент коду, що містить множину команд програми.

Кожен з операторів програми може бути пов'язаний з декількома елементами модифікованого простору: даними (програмними змінними) і командами.

Обчислювальний маршрут при виконанні програми показаний у вигляді направленої пунктирної лінії в сегменті коду. Таким чином виконання програми спричиняє до відтворення семантичних відображень, які відповідають програмним командам. Семантичну коректність програми можна перевірити на підставі аксіоматики, яка буде наведена далі, у так званих контрольних точках, в яких здійснюються підключення механізму вимірювання інваріантів ПР для перевірки обчислювального маршруту за критерієм збереження інваріантів.

2.4 Семантичний простір

Оцінка якості програмного забезпечення за допомогою семантичних інваріантів передбачає:

– формування семантичної моделі ПР, що розглядає семантичне відображення, фактично реалізоване ПР у семантичному просторі, на всіх проектних рівнях в адресному просторі ПР;

– інтерпретацію семантичних відображень і оцінювання семантичних векторів фактично реалізованих у ПР відображень.

Однією з найважливіших характеристик якості ПР є функціональність. Класичні методи верифікації побудовані на перевірці функціональності ПР в адресно-часовому просторі. Пропонується оцінювати функціональність сукупністю коректних семантичних векторів семантичних відображень, фактично реалізованих у ПР, інваріантних до часової складової, що дозволить

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			44

проконтролювати необхідну умову функціональності ПР – збереження семантичних інваріантів за допомогою статичного аналізу програмного коду під час незалежної верифікації.

Концепція, що є основою методу незалежної верифікації, полягає у розгляді характеристик функціональності ІУС у семантичному просторі (СП). Основу семантичного простору становить семантичний базис – n (ортогональних) векторів одиничної довжини, кожний з яких відповідає одній з головних одиниць обраної системи одиниць. Наприклад, для міжнародної системи одиниць СІ базис має дев'ять ортогональних векторів: що мають такі значення та розмірність: Довжина [метр]; Маса [кілограм]; Час [секунда]; Сила електричного струму [Ампер]; Термодинамічна температура [Кельвін]; Сила світла [кандела]; Кількість речовини [моль]; Площинний кут [радіан]; Тілесний кут [стерадіан].

Наприклад, у базисі {час, довжина, маса} розташування механічних одиниць наведено на рисунку 2.3.

Семантичний простір – n -вимірний метричний простір, що складається з множини X елементів, кожний з яких відповідає деякому фізичному типу, що характеризується відстанню – дійсною функцією $\rho(x,y)$, визначеною для будь-яких пар (x,y) з X .

Кожному з елементів СП відповідає семантичний вектор (СВ), який проведено з початку координат. Координатами вектора є розмірність фізичного типу в обраній системі одиниць. Вся сукупність СВ S утворює семантичний простір. СП є лінійним (афінним) простором над полем СВ, тому що існує правило складання, яке дозволяє кожній парі елементів $X, Y \in S$ знайти відповідний елемент $Z \in S$, який називається сумою, та існує правило множення на число, що дозволяє для кожного $X \in S$ та $\lambda \in S$ знайти елемент $U \in S$ – добуток елементу X на число λ , що позначається λX .

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			45

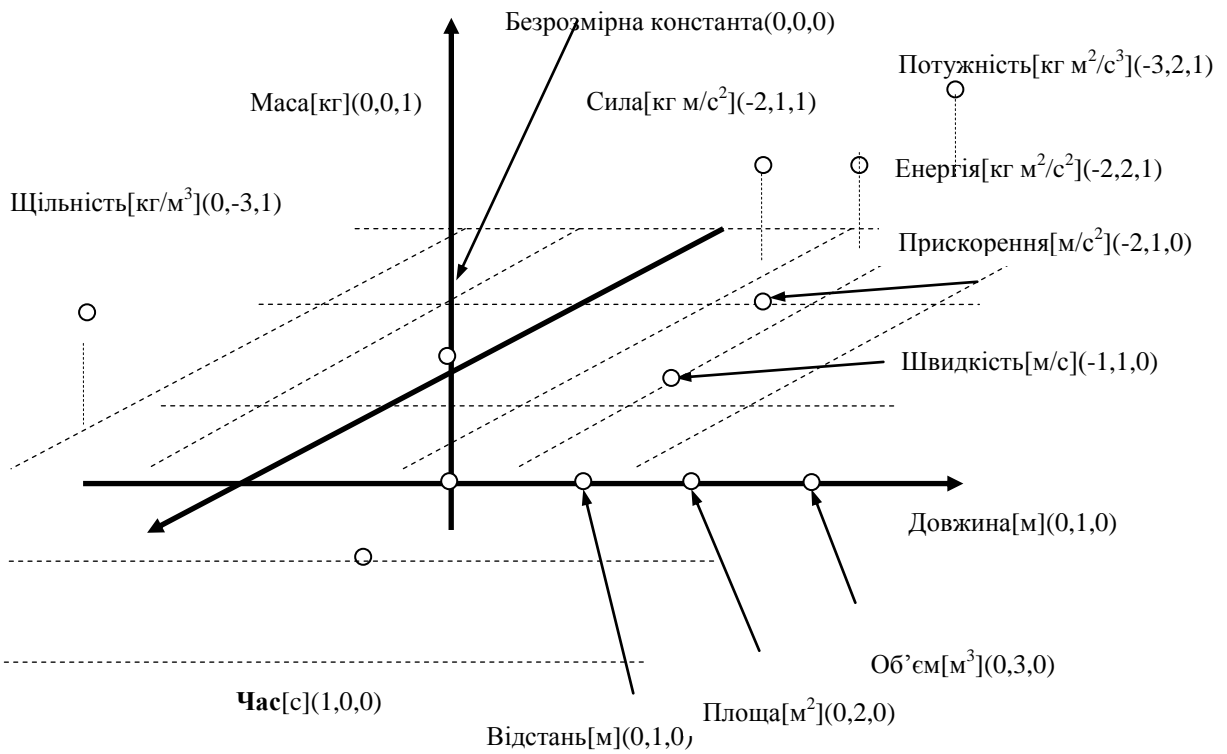


Рисунок 2.3 - Елементи семантичного простору у базисі {Час, Довжина, Маса}

Семантики всіх практично важливих фізичних величин розташовуватимуться в межах гіперкуба, обмеженого інтервалом $(-7,7)$. Операції над елементами СП виконуються за законами векторної алгебри, а семантичний простір є лінійним і векторним.

Основна властивість СП полягає в тому, що його елементами є фізичні типи, які можна порівнювати, множити на константу, складати і віднімати. При цьому утворюються нові фізичні типи. Так, при складанні двох СВ, що відповідають довжині, виникає СВ площі, а при складанні СВ довжини і площі – СВ об'єму. У свою чергу, множення СВ довжини на 2 і 3 також дає відповідно СВ площі і об'єму тощо. Існує взаємоднозначна відповідність між програмними операціями у виразах, що містять фізично типізовані змінні (операторним відображенням) і операціями над елементами СП (семантичним відображенням), яка наведена у таблиці 2.1 та відтворює семантичну алгебру.

Семантичний простір, є лінійним метричним простором, базисом якого є основні одиниці вибраної системи одиниць, а елементами – розмірності

фізичних величин. Тому ж самому елементу простору можуть відповідати декілька програмних змінних, що мають одну розмірність (семантику).

Таблиця 2.1 - Аксиоматика семантичного відображення

№	Операторна конструкція	Семантичне відображення	Умова семантичної коректності
1	$A=B$	$U(s)=A(s)$	$A(s)=B(s)$ Збіг семантичних векторів
2	$A\pm B$		
3	$A > B$		
4	$A < B$		
3	$A*B$	$U(s)=A(s)+B(s)$	Завжди
4	A/B	$U(s)=A(s)-B(s)$	
5	A^N	$U(s)=NA(s)$	$N(s)=0$ – відсутність розмірності показника степеня

Для системи СІ базис семантичного простору включає: метр; кілограм; секунду; Ампер; Кельвін; канделу; моль; радіан; стерadian. На осях координат відкладаються показники степеня. У зв'язку з тим, що всі фізичні типи в системі СІ мають розмірності, що виражаються цілочисельними показниками степенів базових одиниць, елементи семантичного простору також матимуть цілочисельні координати, а сам простір буде дискретним.

Лінійність СП дозволяє розглядати перетворення семантики, здійснювані при виконанні програмного коду, як лінійні відображення. Наприклад, оператору $U = X * Y$ відповідає семантичне відображення $U(s) = X(s) + Y(s)$, де $X(s)$ $Y(s)$ – семантичні вектори співмножників; оператору $U = \frac{X}{Y}$ відповідає семантичне відображення $U(s) = X(s) - Y(s)$ (різниця розмірностей діленого та дільника); оператору $U = X^Y$ (піднесення до цілого степеня) відповідає $U(s) = Y * X(s)$, якщо $Y(s) = 0$. Таким чином, операції, що виконуються над програмними змінними під час роботи програми, фактично є операціями над

									Арк.
									47
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

векторами в лінійному метричному просторі, а відображення семантики, що здійснюються під час виконання – лінійними відображеннями.

Результуюче семантичне відображення програми не залежить від порядку семантичних відображень програмних операторів.

Припустимо, що X та Y – семантичні простори над полем скалярів K . Відомо, що відображення $U: X \rightarrow Y$ називається лінійним відображенням або лінійним оператором, якщо $U(\lambda x + \mu y) = \lambda U(x) + \mu U(y)$ для всіх $\lambda, \mu \in K$ та $x, y \in X$. У випадку семантичних відображень таким відображенням буде піднесення до степеня. Доведемо далі, що множину всіх лінійних відображень з $U: X \rightarrow Y - L(X, Y)$ можна розглядати як векторний простір, за умови визначення алгебричних операцій таким чином.

Припустимо, що $U_1, U_2 \in L(X, Y)$. За визначенням $U = U_1 + U_2$ є оператор відображення $U: X \rightarrow Y: U(x) = U_1(x) + U_2(x)$, де $x \in X$.

Очевидно, що $U \in L(X, Y)$. Якщо $U \in L(X, Y)$ для $\lambda \in K$, то відображення $\tilde{U} = \lambda U$ визначається так: $\tilde{U}(x) = \lambda U(x)$, де $x \in X$.

Для будь-якого семантичного відображення $U(x) \in L(X, Y)$ маємо $U(0) = 0$ та $U(-x) = -U(x)$, де $x \in X$. Семантичному відображенню $U(0) = 0$ відповідає таке відображення, що нульовому елементу семантичного простору X ставить у відповідність нульовий елемент семантичного простору Y .

Лінійність векторного простору $L(X, Y)$, елементами якого є елементарні семантичні вектори, що відповідають аксіомам семантичної алгебри, дозволяє використовувати алгебричний контроль семантичної коректності ПР без виконання за штатною схемою [15].

2.5 Розширений семантичний простір

Подальшим розвитком семантичного вектора є семантичне число, яке, на відміну від СВ, містить додаткову кількісну характеристику: $X = \{X_0, X_1, X_2, X_3 \dots X_N\}$, де X_0 – кількісна характеристика (реальне число), а $X_1, X_2, X_3 \dots X_N$ – семантичний вектор, який відповідає фізичному типу результату в семантичному просторі.

Семантичні числа дозволяють отримувати разом із кількісним результатом також і якісний – його фізичний тип. Аксиоматика розширеної семантичної алгебри, що визначає операції над семантичними числами, подана у таблиці 2.2.

Таблиця 2.2 - Операції над семантичними числами

Операція	Мат. запис	Результат	Умова виконання
Складання	$Z = X + Y$	$Z_0 = X_0 + Y_0$	$X_i = Y_i, \forall i \neq 0$
Віднімання	$Z = X - Y$	$Z_0 = X_0 - Y_0$	$X_i = Y_i, \forall i \neq 0$
Множення	$Z = X * Y$	$Z_0 = X_0 * Y_0$ $Z_i = X_i + Y_i$	–
Ділення	$Z = X / Y$	$Z_0 = X_0 / Y_0$ $Z_i = X_i - Y_i$	–
Піднесення до степеня	$Z = X^n$	$Z_0 = X_0^n$ $Z_i = X_i * n$	–
Порівняння	$X \text{ or } Y$	$X \text{ or } Y$, де or – операція порівняння	$X_i = Y_i, \forall i \neq 0$

Визначимо згідно з класичною алгеброю закони множення для семантичних чисел:

- 1) закон асоціативності: $a(bc) = (ab)c$;
- 2) закон комутативності: $ab = ba$;
- 3) існує одиниця, тобто елемент e , причому $ea = a$ для всіх a . У множині семантичних чисел такий одиничний елемент буде мати значення: $e = \{1, 0, 0, \dots, 0\}$;

4) для кожного елемента a існує один обернений елемент a^{-1} із властивістю $a^{-1}a = e$; $a^{-1} = \left\{ \frac{1}{a_0}, -a_1, -a_2, \dots, -a_n \right\}$, за умови що $a_0 \neq 0$.

Стосовно операції множення семантичні числа утворюють комутативну групу. Розглянемо властивості сумування семантичних чисел:

1) існує нескінченна множина „нульових” елементів із властивостями: $0 + a = a$. Семантичний нуль – елемент, що має нульову кількісну характеристику та еквівалентний семантичний вектор $\{0, a_1, a_2, \dots, a_n\}$. Такий елемент, що існує для кожного із семантичних векторів, не змінює значення семантичного числа при складанні з ним;

2) існує обернений елемент $-a$, із властивістю $-a + a = 0$, причому $-a = \{-a_0, a_1, a_2, \dots, a_n\}$;

3) закон асоціативності: $a + (b + c) = (a + b) + c$, якщо $a_i = b_i = c_i$ де $i \neq 0$;

4) закон комутативності: $a + b = b + a$ за умови $a_i = b_i$ де $i \neq 0$;

5) можливість розв’язання рівняння $a + x = c$, за умови, $a_i = x_i = c_i$ $i \neq 0$.

Що стосується сумування, то через наявність умови виконання операції семантичні числа адитивної групи не утворюють;

б) закон дистрибутивності семантичних чисел: $a(b + c) = ab + ac$;
 $(b + c)a = ba + ca$.

Реалізація операцій за аксіоматикою розширеної семантичної алгебри дозволить під час динамічного тестування також контролювати збереження семантичних програмних інваріантів, що значно підвищить не тільки ефективність тестів та верифікації, але й надійність програмного забезпечення інформаційно-управляючих систем.

Надалі побудовану систему з подвійною композицією будемо називати семантичним кільцем, а сукупність елементів $\{X\}$ – розширеним семантичним простором.

Екземпляри класу X відповідають елементам необмеженого розширеного семантичного простору, що є метричним з метрикою $\rho(a, b)$, такою, що:

									Арк.
									50
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

$$\rho(a, a) = 0;$$

$$\rho(a, b) = \rho(b, a);$$

$$\rho(a, c) \leq \rho(a, b) + \rho(b, c).$$

Будемо вважати розширений семантичний простір евклідовим з метрикою: $\rho(a, b) = \sqrt{\sum (a_i - b_i)^2}$

За таким визначенням метрики простору, будь-які два результати, що мають різні семантики, будуть вважатися різними, що дозволить верифікувати ПР не тільки за збігом значень (їх приналежністю ϵ -довкіллю еталонних значень), і за фізичним типом, що визначається збігом семантичних складових.

2.6 Необхідна умова коректності програмних ресурсів

Сформулюємо необхідну умови коректності ПР: ПР семантично коректні, якщо на всіх рівнях зберігається семантика програмних змінних. Достатня умова коректності досягається виконанням всіх функцій та повною відсутністю дефектів програмного забезпечення.

Запропонована необхідна умова семантичної коректності більш слабка та не може забезпечити 100% відсутність програмних дефектів. Але, в умовах ресурсних обмежень, перевірка необхідної умови коректності програмного забезпечення шляхом формальної верифікації, з використанням контролю коректності семантичних перетворень, за допомогою АТД або методів СА, дозволить виявити програмні дефекти (ПД), не виявлені розробником, під час ретельного тестування та верифікації.

Розглядання семантики дозволяє використати програмний інваріант – збереження семантики програмних даних для формальної верифікації програмного забезпечення. Зміна семантичних векторів даних буде свідчити про некоректність програми. Наприклад, спроба присвоєння змінній, тип якої „маса”, деякого значення, що відповідає „часу”.

									Арк.
									51
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

Протилежне не вірно, тобто факт збереження семантики не може свідчити про вірність результату. Таким чином, для кожної адитивної операції повинно виконуватися: $\sum_i (L_i - R_i)^2 = 0, \forall i \neq 0$, де L, R – семантичні вектори лівого та правого операндів.

Для коректно побудованої програми семантика очікуваного результату збігається з семантикою фактичного результату. Тому необхідною умовою семантичної коректності буде: $\sum_j \sum_i (L_i - R_i)^2 = 0, \forall i \neq 0$, де i – номер компоненти семантичного вектора; j – номер програмної конструкції.

В той же час достатньою умовою існування програмних дефектів є $\sum_j \sum_i (L_i - R_i)^2 \neq 0, \forall i \neq 0$.

Перевірка останньої умови дозволяє з точністю до виразу визначити порушення семантики і таким чином – місцезнаходження програмного дефекту.

Формальна семантична верифікація дозволить виявити такі класи програмних дефектів без виконання програми в реальних умовах:

- 1) використання невірних ідентифікаторів, типів операндів, операцій в арифметичних та логічних виразах;
- 2) семантичні невідповідності формально-фактичних параметрів;
- 3) використання невірних адрес операндів;
- 4) використання невірних операцій;
- 5) порушення ходу обчислювального процесу, пов'язане з пропуском деяких програмних модулів або виконанням зайвих програмних модулів.

2.7 Концептуальна модель програмних ресурсів

Архітектура ПР розглядається як ієрархія взаємозв'язаних об'єктів:

- інформаційно управляюча система (ІУС) в цілому;
- ПТК, що входять до складу ІУС;
- задачі, що виконують ПТК;

										Арк.
										52
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ					

- окремі функції, що входять до складу задач;
- оператори програмного коду, що реалізують функції.

Такий підхід зручний для подання взаємозв'язків на всіх рівнях оцінюваного програмного проекту і спрощує реалізацію і супровід ER-моделі об'єкта експертизи існуючими реляційними БД (рисунок. 2.4).

Як вже відзначалося, функціональність ПР розглядається як сукупність відображень області визначення на області допустимих значень у семантичному просторі.

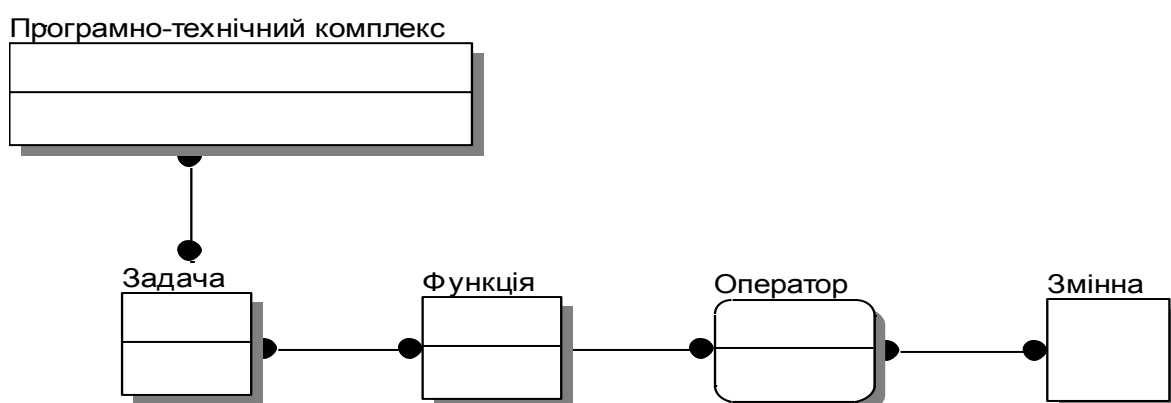


Рисунок 2.4 - ER-модель програмних ресурсів

Загальна форма відображень для кожного з рівнів моделі об'єкта експертизи може бути показана у векторному вигляді:

$$SY_j = F_j(SX_i), \quad (2.1)$$

де SY_j – множина вихідних СВ;

SX_i – множина вхідних СВ;

F_j – векторне семантичне відображення множини вхідних СВ на вихідний СВ j -параметра.

Узагальнення на всі рівні проекту дозволяє відобразити функціональність ПР в матричному вигляді:

										Арк.
										53
Змн.	Арк.	№ докум.	Підпис	Дата						

$$SY_{jk} = F_{jk}(SX_{ik}), \quad (2.2)$$

де SY_{jk} – вихідна семантична матриця, що містить СВ всіх j -вихідних параметрів, k -проектних рівней;

SX_{ik} – вхідна семантична матриця, що містить вхідні i -СВ для всіх k -проектних рівней;

F_{jk} – функціональна матриця семантичного відображення множини вхідних i -параметрів на j -вихідний параметр для k -проектних рівнів, елементи якої – семантичні відображення областей визначення на області допустимих значень для k -рівня.

Через лінійність семантичного відображення матриця F_{jk} також лінійна, що дозволяє запропонувати незалежну верифікацію ПР ІУС шляхом статичного аналізу ПР з контролем його семантичної коректності.

На відміну від традиційної верифікації, що заснована на виконанні сукупності тестів, які покривають деякі набори вхідних даних, модель, що пропонується, забезпечує контроль коректності для всіх змінних на всіх рівнях проекту, що задаються функціональною матрицею F_{jk} . При цьому забезпечується повне покриття коду.

Критеріями оцінки ПР з використанням метрик семантики є коректність семантичних відображень, яка визначається у збігу СВ операндів адитивних операцій та операції присвоєння.

Результати семантичної верифікації, що стосуються окремих операторів програмного коду, програмних модулів, алгоритмів, функцій, задач, ПТК та оперативної бази даних для зберігання телеметричної інформації, що входять в семантичну модель ІУС АЕС, співвідносяться між собою згідно з концептуальною моделлю об'єкта експертизи, поданою в Додатку Б.

									Арк.
									54
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

2.8 Семантична модель дефектів операндів

Для оцінки ефективності незалежної верифікації, заснованої на контролі коректності семантичних відображень, необхідне знання перевіряючої спроможності методу – умовної ймовірності виявлення програмних дефектів (ПД) за умови його існування.

Розглянемо далі вплив семантичного і операційного спектрів на ефективність пропонованого методу незалежної верифікації для реального ПР ІУС, скориставшись ймовірнісними семантичними моделями дефектів.

Розглянемо найпростішу семантичну модель ПД, що припускає, що в кодї можуть спотворюватися тільки операнди. Ймовірнісний граф подій такої моделі дефектів подано на рисунку 2.5.

Знайдемо ефективність семантичного методу незалежної верифікації як умовну ймовірність виявлення ПД за умови існування самого ПД.

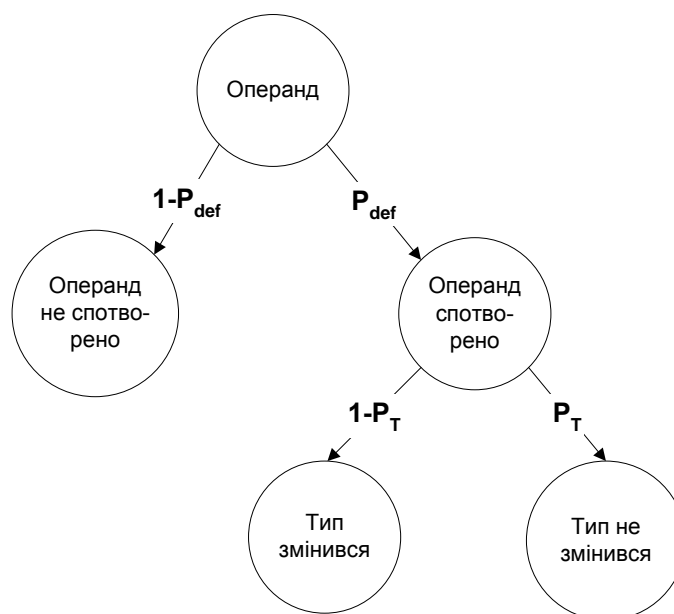


Рисунок 2.5 – Ймовірнісний граф подій моделі дефектів операндів

Запропонована модель обмежує всі дефекти ПР тільки дефектами операндів. Перебачається, що спотворення операндів може відбуватися з ймовірністю P_{def} , а відсутність дефектів операндів – з ймовірністю $P_{\overline{def}} = 1 - P_{def}$.

У разі дефекту спотворення може бути розпізнаним, якщо воно змінює

									Арк.
									55
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

семантику операнду. У разі коли фізична розмірність помилкового операнду збігається з потрібною розмірністю, що має ймовірність P_T (ймовірність збігу фізичної розмірності операнду при спотворенні), дефект не розпізнається.

Через те, що нас цікавить тільки випадок існування ПД, позначимо ефективність семантичної моделі:

$$\eta = \frac{P_f}{P_f + P_{\bar{f}}} , \quad (2.3)$$

де $P_f, P_{\bar{f}}$ – ймовірності виявлення та невиявлення ПД.

Надалі будемо припускати, що програма з ймовірністю P_{def} може містити ПД, тоді ймовірність відсутності ПД. – P_n може бути знайдена:

$$P_n = 1 - P_{def} . \quad (2.4)$$

На підставі моделі, що пропонується, знайдемо ймовірність виявлення ПД, котрий призводить до зміни фізичного типу операнду:

$$P_f = P_{def}(1 - P_T) . \quad (2.5)$$

Невиявлення ПД можливе у тому випадку, коли фізичний тип операнду не змінюється. Ймовірність такого випадку визначається так:

$$P_{\bar{f}} = P_{def}P_T . \quad (2.6)$$

Ці ймовірності є функціями відносного розподілу програмних даних за фізичними типами (семантичного спектру ПР).

Для перевірки коректності семантичної моделі ПД скористуємося тим, що сума ймовірностей несумісних подій, що утворюють повну групу: дефект

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

відсутній; дефект присутній та виявлений; дефект присутній та невиявлений, повинна дорівнювати 1:

$$P_n + P_f + P_{\bar{f}} = 1, \quad (2.7)$$

де P_n – ймовірність відсутності ПД;

P_f – ймовірність виявлення ПД;

$P_{\bar{f}}$ – ймовірність невиявлення ПД.

Підставимо (2.5) – (2.6) у (2.7), результатом буде тотожність:

$$1 - P_{def} + P_{def}(1 - P_T) + P_{def}P_T = 1, \quad (2.8)$$

що доводить коректність моделі.

Підставимо (2.5), (2.6) у (2.3) та отримаємо, що для семантичної моделі ПД операндів ефективність визначається тільки семантичним спектром та не залежить від ймовірності існування програмних дефектів у ПР P_{def} :

$$\eta = 1 - P_T. \quad (2.9)$$

Розглянемо найпростіший випадок – рівномірний розподіл змінних за різними фізичними типами, тобто рівномірний семантичний спектр. Для 10 семантичних різних фізичних типів даних, кожний з яких представлений однаковою обсягом даних, ймовірність збігу типів при спотворенні операнду становитиме 0.1, а ефективність моделі – 90%.

У випадку нерівномірного семантичного спектру ймовірність розпізнавання дефектів визначатиметься складовими семантичного спектру, що потребує статистичного аналізу коду.

Семантичний спектр зазначеного ПР містить 11 різних фізичних типів даних, які займають від 0.2% до 37%. Перевіряючи спроможність для

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

семантичної моделі дефектів операндів для зазначеного коду буде у діапазоні 63...99.75%. Середня ефективність може бути обчислена за формулою:

$$\eta_{cp} = \sum_{i=1}^n 1 - P_{Ti} P_{Ti}, \quad (2.10)$$

де P_{Ti} – частка програмних даних відповідного типу.

У нашому випадку для розподілу типів даних, наведеного у табл. 2.4, середня перевіряюча спроможність становитиме близько 77%.

Формула (2.10) разом з додатковою вимогою:

$$\sum_{i=1}^n P_{Ti} = 1 \quad (2.11)$$

дозволяє знайти залежність середньої ефективності від кількості різних типів при припущенні їх рівномірного розподілу: $\eta_{cp} = n \left(1 - \frac{1}{n}\right) \frac{1}{n}$; $n \frac{1}{n} = 1$.

Звідки

$$\eta_{cp} = \left(1 - \frac{1}{n}\right) \frac{1}{n}, \quad (2.12)$$

де n – кількість різних фізичних типів, що використано у програмі.

Границя цього виразу при необмеженому зростанні кількості фізичних типів програмних змінних дорівнює 1.

									Арк.
									58
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

3 РОЗРОБКА ТА ОПИС ПРОГРАМНОГО СЕРВІСУ ІНТЕРАКТИВНОГО КЕРУВАННЯ ВЕБ-РЕСУРСАМИ

3.1 Структура розробленого програмного забезпечення інтерактивного керування веб-ресурсами

Розробка системи інтерактивного керування веб-ресурсами полягає, щоб кінцевий користувач вносив зміни та доповнення на сайт через зручний редактор, без спеціальних вмінь та навичок і весь процес займав менше затрат і часу. Для цього було розроблено зручну структуру керування веб-ресурсом, представлену в Додатку В.

Принцип роботи керування веб-ресурсами ґрунтується на поділі контенту (змісту) і дизайну (оформлення) сайту. Звичайно дизайн сайту змінюється рідко, тоді як зміни контенту можуть відбуватися не тільки щодня, але й навіть щогодини. Тому у своїй роботі система керування веб-ресурсами використовує так звані шаблони - спеціальні "порожні" заготовки сторінок, у яких дизайн сайту вже прописаний і залишилося лише наповнити їх інформацією. Користувачеві досить скористатися спеціальним WYSIWYG-редактором який зображений на рисунку 3.1.

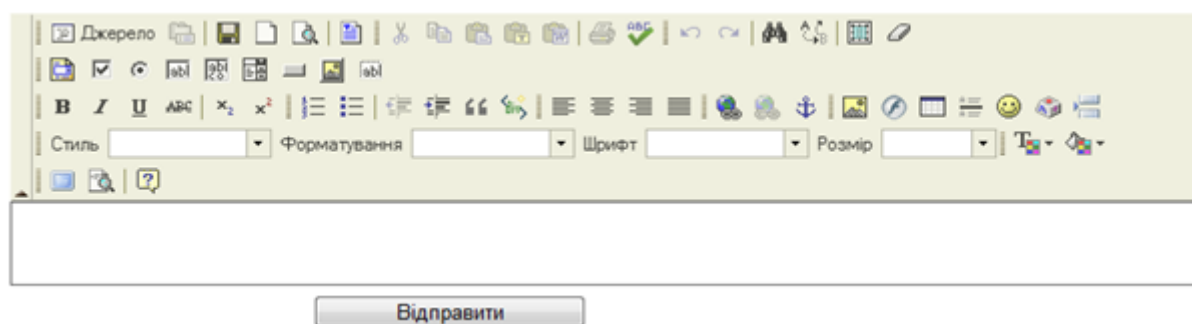


Рисунок 3.1 - WYSIWYG-редактор

Цей редактор за зовнішнім виглядом дуже схожий на звичні текстові редактори офісних додатків, тому користувачеві не особливо важко освоїти його. А наявність у системі великої кількості готових шаблонів дає можливість вибрати потрібний дизайн буквально в лічені хвилини.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Також основними перевагами розробленої системи інтерактивного керування веб-ресурсами є:

- зручна установка;
- зручне управління структурою веб-ресурса;
- створення і видалення розділів і підрозділів, пунктів меню вебресурса;
- внесення тексту та малюнків на веб-ресурс;
- редагування сторінок та мета-тегів веб-ресурса;
- редагування текстів сторінок веб-ресурса;
- редагування шаблону із адмінвідділу;
- наявність seo оптимізатора з унікальними title, description, keywords для сторінок.

Розберемо детальніше переваги.

3.2 Функціональність розробленого програмного забезпечення

3.2.1 Установка програмного забезпечення

Програмний сервіс був розроблений для зручної та швидкої установки, не навантажує сервер та займає дуже мало дискового об'єму. Розглянемо послідовність установки даного програмного сервісу (рисунок 3.2).

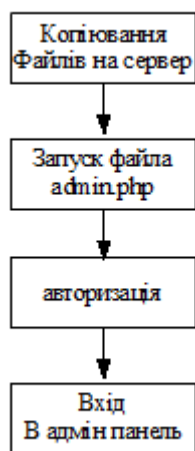


Рисунок 3.2 - Послідовність установки розробленого програмного сервісу.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

Щоб розпочати працювати з даним програмним сервісом, потрібно скопіювати каталоги з файлами які зображені на рисунку 3.3.

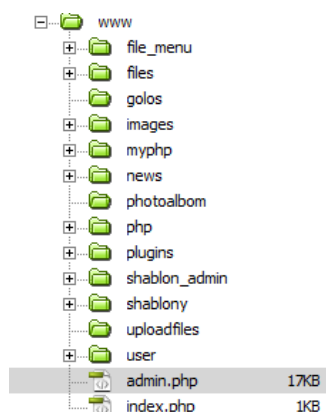


Рисунок 3.3 - Дерево каталогів системи управління веб-ресурсами.

Після цього запускаємо файл `admin.php` в браузері і попадаємо на сторінку авторизації (рисунок 3.4), логін та пароль стоїть по замовченню (`admin`). Після того в адміністраторській панелі заходимо в пункт меню зміну паролю і змінюємо на інший для безпеки даних.

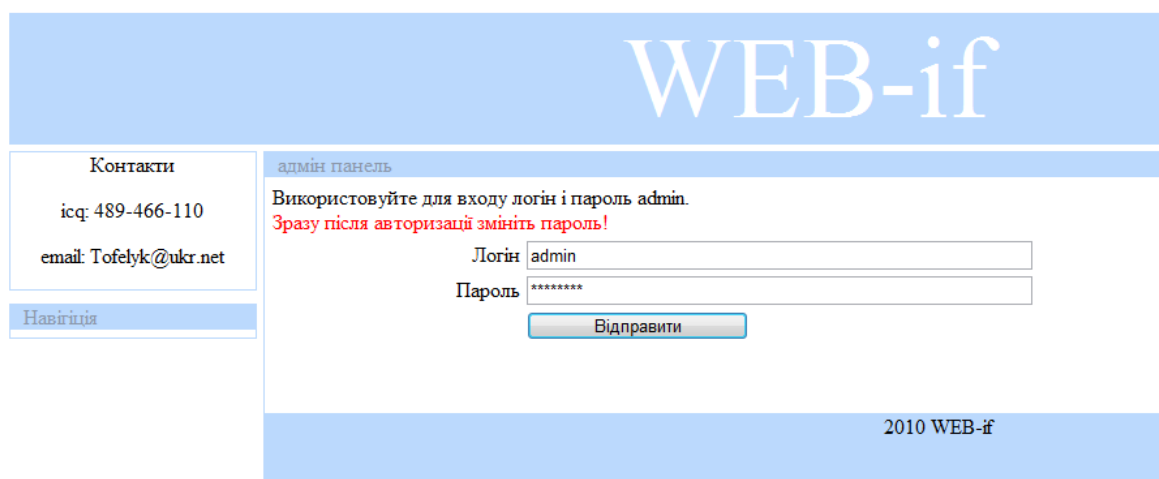


Рисунок 3.4 - Сторінка авторизації

Після введення логіну та паролю відкривається адмін панель, в якій вже можна почати вносити зміни в сайт (рисунок 3.5).

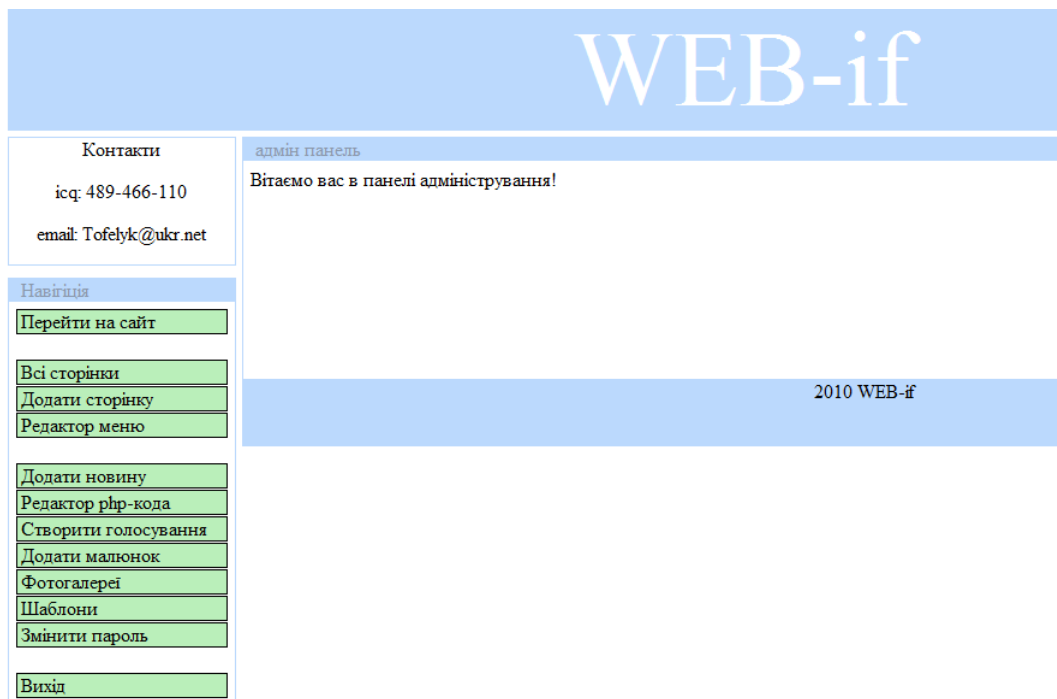


Рисунок 3.5 - Адміністраторська панель

3.2.2 Принцип змінення та налаштування шаблону веб-ресурса

Однією із переваг системи є зручне і швидке змінення шаблону веб-ресурса.

До програмного забезпечення інтерактивного керування веб-ресурсами було розроблено алгоритм, за допомогою якого користувач даного програмного сервісу може з легкістю і без спеціальних навичок змінити шаблон сайту. До системи легко прикріплюються шаблони, яких в Інтернеті дуже багато, і частина з них абсолютно безкоштовна. А так як html-код та код PHP розділені, то й керувати вмістом і дизайном дуже зручно і просто також в систему були розроблені редактори редагування як HTML-коду так і CSS стилів.

Розроблений принцип змінення шаблону працює по алгоритму, схема якого представлена на рисунку 3.6.

Шаблон знаходиться в папці shablon. Всі зображення, необхідні для роботи шаблону, повинні знаходитися в папці imgweb_if. Файл стилів так само повинен знаходитися в папці imgweb_if.

Файл шаблону-index.html. У ньому зберігається html-код шаблону. У код вставляються спеціальні мітки:

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62



Рисунок 3.6 - Принцип змінення шаблону.

- [Title] - title сторінки;
- [Description] - meta-тег description сторінки;
- [Keywords] - meta-тег keywords сторінки;
- [Zagolovok] - заголовок сторінки;
- [Menu] - меню сайту;
- [Tormenu] - коротке меню сайту;
- [Body] - зміст сторінки;
- [Php] - вставка коду php;
- [News] – новини;
- [Golos] – опитування.

Такий принцип змінення шаблону було розроблений для того, щоб можна було скористатися будь-яким шаблоном, який розроблений з допомогою технологій HTML і CSS.

Розставивши всі мітки в своєму шаблоні, можна закачати файли в папку shablon.

Після цього треба зайти в пункт навігації шаблони і вибрати даний шаблон із переліку, який зображений на рисунку 3.7.

							ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата				63



Рисунок 3.7 - Сторінка вибору шаблону для сайту

Дану функцію вибору шаблону обробляє код:

```

if (isset($_GET["shablony"]))
$ret=shabl_menu2()
if (isset($_GET["fname"]))
if (file_exists("user/shablony"))
{
include('user/shablony');
$handle = fopen("user/shablony", "w");
fwrite($handle,'<? $shablonyname="'.$_GET["fname"]."'> ?>');
fclose($handle);
$shablonyname=$_GET["fname"];
}
$ret.="<font color=red>Вибір шаблону ".$shablonyname."</font><br><br>";
$ret.="<b>Всі шаблони</b><br>";
if ($hdir = opendir('shablony/')) {
$i=0;$bodyadd="";

```

```

while (false !== ($file = readdir($hdir))) {
if ($file != "." && $file != ".." && $file != "error_log" && $file!="shablon") {
$bodyadd.="[<a href=shablony/'".$file."/index.html target=frame1 title='Нажми для
предпросмотра'>".$file."</a>] - <a
href=admin.php?shablony&fname='".$file."><font color=green>Встановити цей
шаблон</font></a><br>";
}
}
closedir($hdir);
}

```

Також в цьому пункті меню можна внести зміни як в шаблоні, так і в стилях за допомогою вбудованих в програмний сервіс редакторів, які зображені на рисунках 3.8 та 3.9.

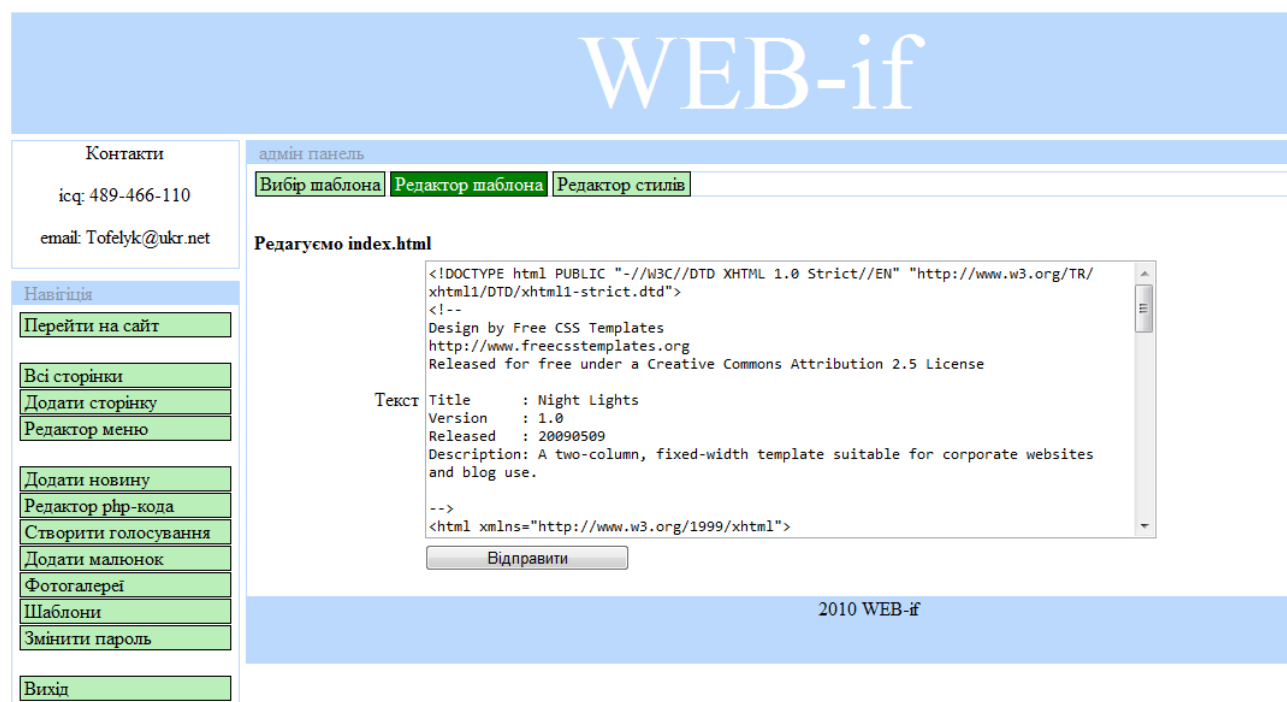


Рисунок 3.8 - Редактор шаблону

Редагуємо style.css

```

/*
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License
*/

* {
    margin: 0;
    padding: 0;
}

body {
    background: #D5DBB5;
    text-align: justify;
    font-family: Arial, Helvetica, sans-serif;
}

```

Текст

Відправити

Рисунок 3.9 - Редактор стилів

3.2.3 Створення нової сторінки

Головна сторінка готова. Переходимо до створення додаткових сторінок. Потрібно натиснути додати розділ в пункті навігації.

Після цього можна заповнювати поля title, keywords, description. Можна залишити порожніми title, keywords, description, заголовок - вони заповняться автоматично даними з головної сторінки, але все-таки краще їх заповнити. Якщо введено title, але не введений заголовок - заголовок буде як title.

Принцип додавання сторінки був розроблений по всім категоріям юзабіліті. Так щоб користувач, який не має спеціальних знань, мав можливість зручно і швидко додати нову або редагувати вже існуючу сторінку.

Розроблений алгоритм додавання сторінки та її редагування зображений в Додатку Г.

Створення нових сторінок представлено на рисунку 3.10 і включає такі операції:

- створення необмеженої кількості сторінок;
- створення розділів сайту;
- створення "вкладених" підрозділів у структурі розділів сайту.

										Арк.
										66
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ					

Розділ – це головний (верхній) об'єкт в основі ієрархії структури вмісту.
 Пізніше в Розділ будуть додані Категорії.

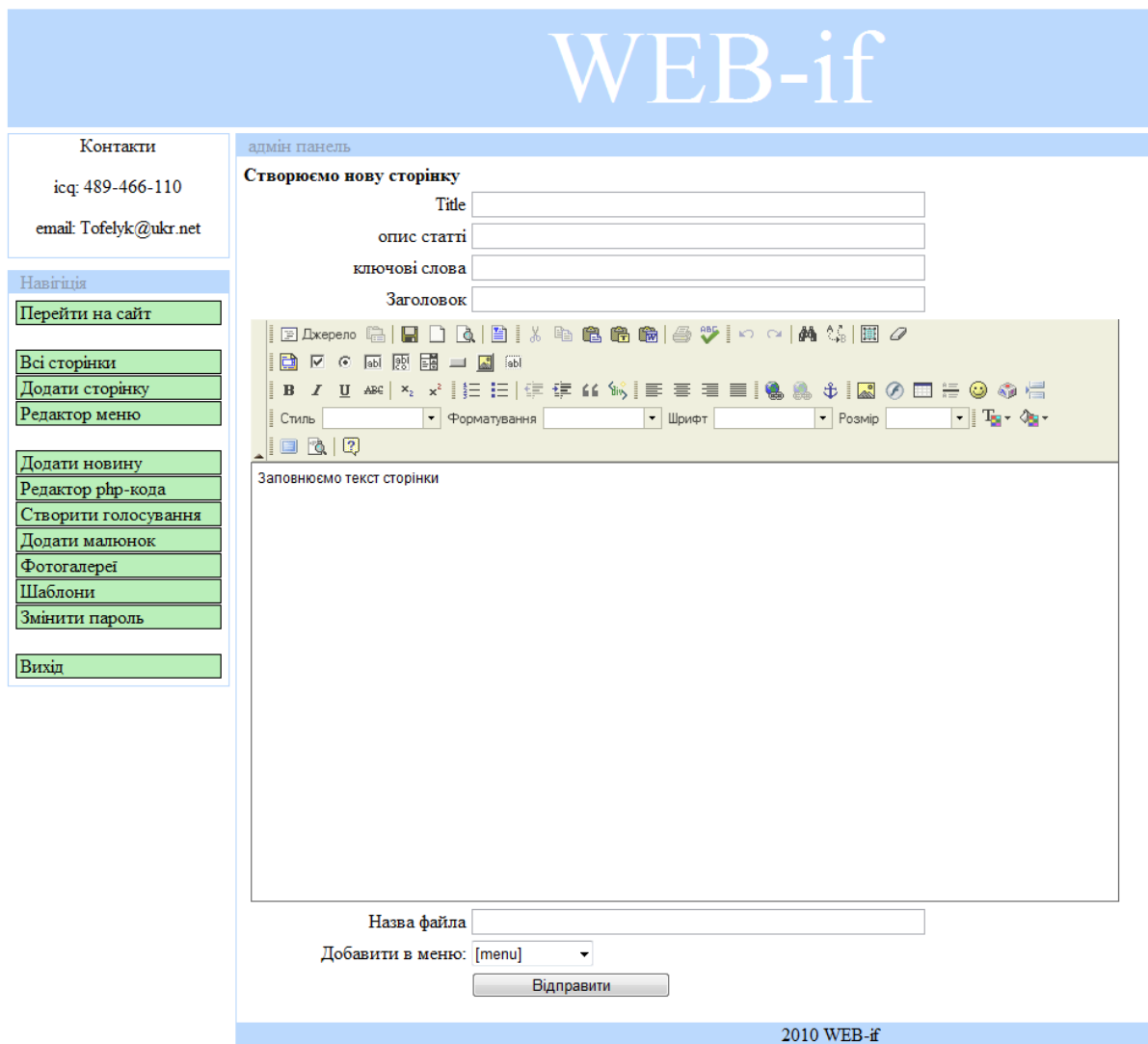


Рисунок 3.10 - Створення нової сторінки

В полі файл даємо назву файла. Якщо не ввести назву файлу, то буде додано автоматично і складатися з цифр. Після заповнення всіх полів треба натиснути Надіслати.

Файл створений, залишилося тільки поставити посилання на нього з редактора меню (рисунок 3.11). Натиснувши посилання редактора меню, можна потрапити на сторінку редагування меню. В таблиці «добавляємо сторінки меню» вже обрано файл, який був створений, а залишилося ввести назву.

										Арк.
										67
Змн.	Арк.	№ докум.	Підпис	Дата						

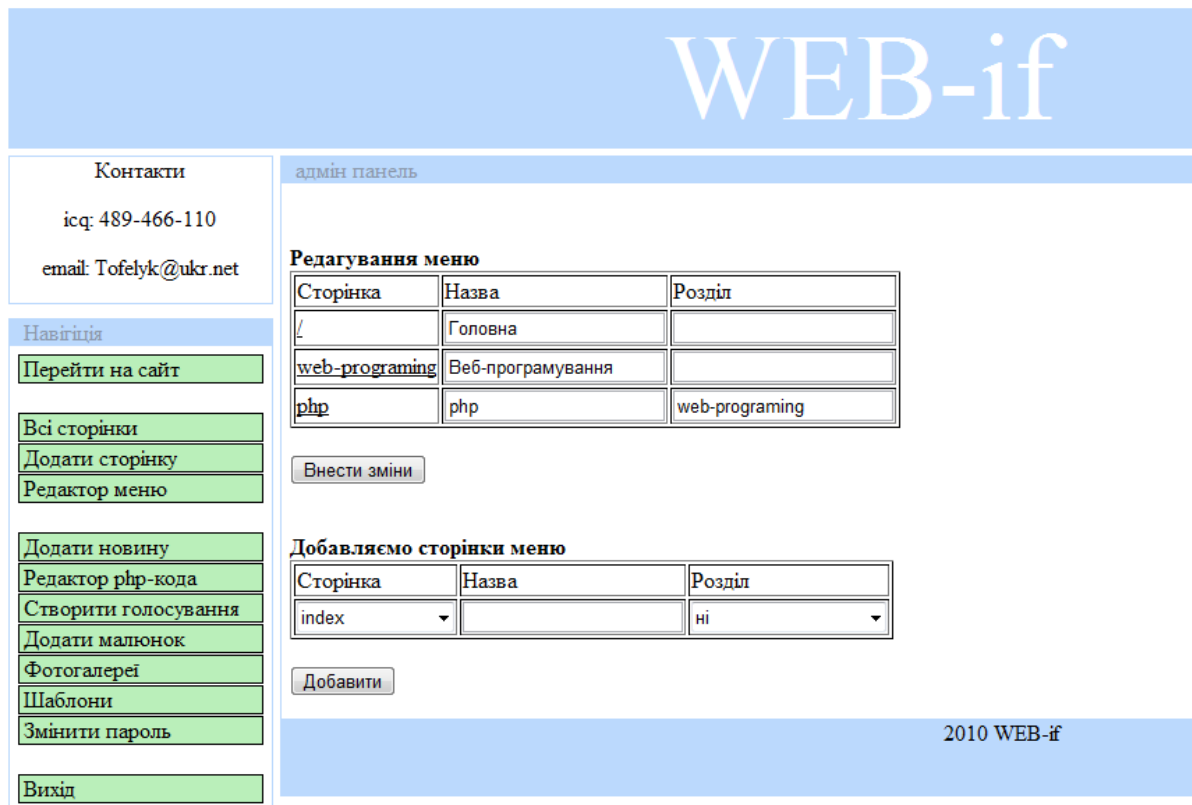


Рисунок 3.11 - Сторінка редагування меню

3.2.4 Додавання та керування статті на веб-ресурсі

Для того щоб додати статтю на веб-ресурс через адмін панель, потрібно перейти в меню навігації до пункту «Додати новину» (рисунки 3.12-3.14).

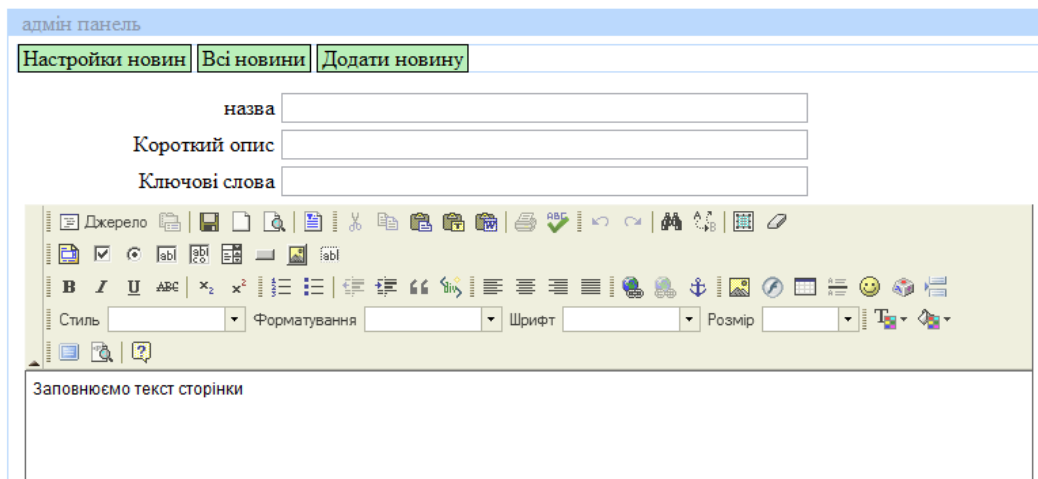


Рисунок 3.12 – Сторінка створення статті



Рисунок 3.13 - Настройки новин

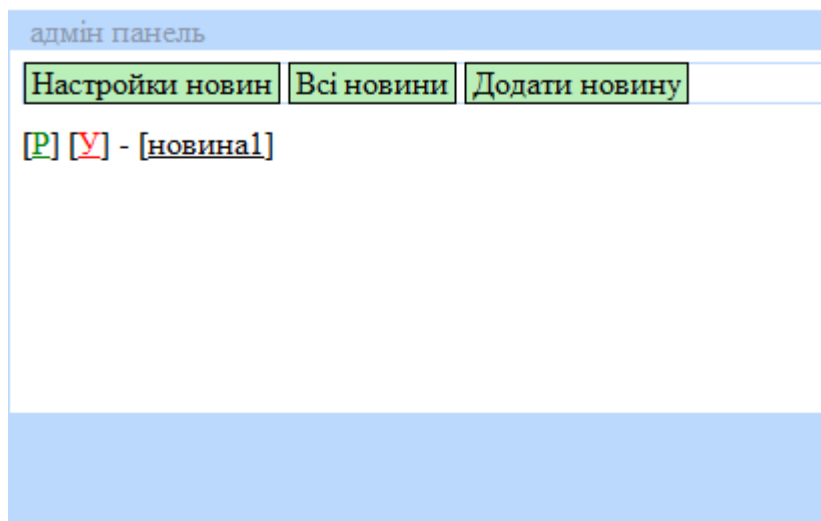


Рисунок 3.14 – Сторінка перегляду всіх новин

3.2.5 Додаткові функції інтерактивного керування веб-ресурсами

Редактор php-коду викликається в тих місцях шаблону, де вставлена мітка [php]. У місці мітки викликається файл code1.php. Там, де знаходиться друга мітка, викликається code2.php. І так можна розмістити 5 позначок (рисунок 3.15).

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			69

адмін панель

- [code1.php] - [Редагувати]
- [code2.php] - [Редагувати]
- [code3.php] - [Редагувати]
- [code4.php] - [Редагувати]
- [code5.php] - [Редагувати]

Рисунок 3.15 – Вибір редагування php коду на сторінці веб-ресурса

На рисунку 3.16 зображено сторінку редагування php коду.

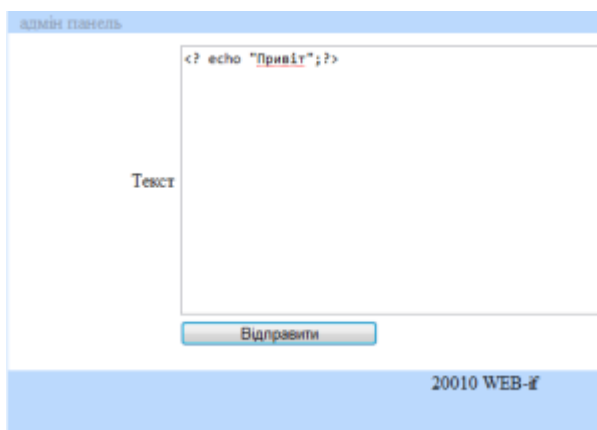


Рисунок 3.16 - Сторінка редагування php-коду

3.3 Тестування програмного сервісу

Розроблене програмне забезпечення інтерактивного керування веб-ресурсами - це програмний комплекс, що дозволяє автоматизувати процес управління як сайтом в цілому, так і сутностями в рамках сайту: макетами сторінок, шаблонами виведення даних, структурою, інформаційним наповненням, користувачами і правами доступу, а також по можливості надає додаткові сервіси: створення голосування, додавати фотогалереї, засоби взаємодії з користувачами і т. д.

З самого початку потрібно зайти в адміністраторську панель, ввести логін та пароль, який стоїть по замовчуванню(admin) (рисунок 3.17). Потім треба зайти в пункт меню шаблони (рисунок 3.18-3.19) і вибрати із списку шаблон, який більше підходить до тематики сайту.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

адмін панель

Використовуйте для входу логін і пароль admin.
Зразу після авторизації змініть пароль!

Логін

Пароль

2010 WEB-іf

Рисунок 3.17 - Сторінка авторизації користувача

Навігіція

Рисунок 3.18 - Панель навігації

<p>Контакти</p> <p>icq: 489-466-110</p> <p>email: Tofelyk@ukr.net</p>	<p>адмін панель</p> <p><input type="button" value="Вибір шаблону"/> <input type="button" value="Редактор шаблону"/> <input type="button" value="Редактор стилів"/></p> <p>Вибір шаблону</p> <p>Всі шаблони</p> <p>[carcar] - Встановити цей шаблон</p> <p>[click_blue] - Встановити цей шаблон</p> <p>[collectiveblue] - Встановити цей шаблон</p> <p>[delfin] - Встановити цей шаблон</p> <p>[emblazoned] - Встановити цей шаблон</p> <p>[megapolis] - Встановити цей шаблон</p> <p>[nightlights] - Встановити цей шаблон</p> <p>[OrangeVillage] - Встановити цей шаблон</p> <p>[presented] - Встановити цей шаблон</p> <p>[zakat] - Встановити цей шаблон</p>
---	---

Навігіція

Рисунок 3.19 - Сторінка вибору шаблону для веб-ресурса

Після цього потрібно зайти в пункт меню «змінити пароль» і змінюємо пароль авторизації для безпеки даних, а тоді перейти на сайт (рисунок 3.20).

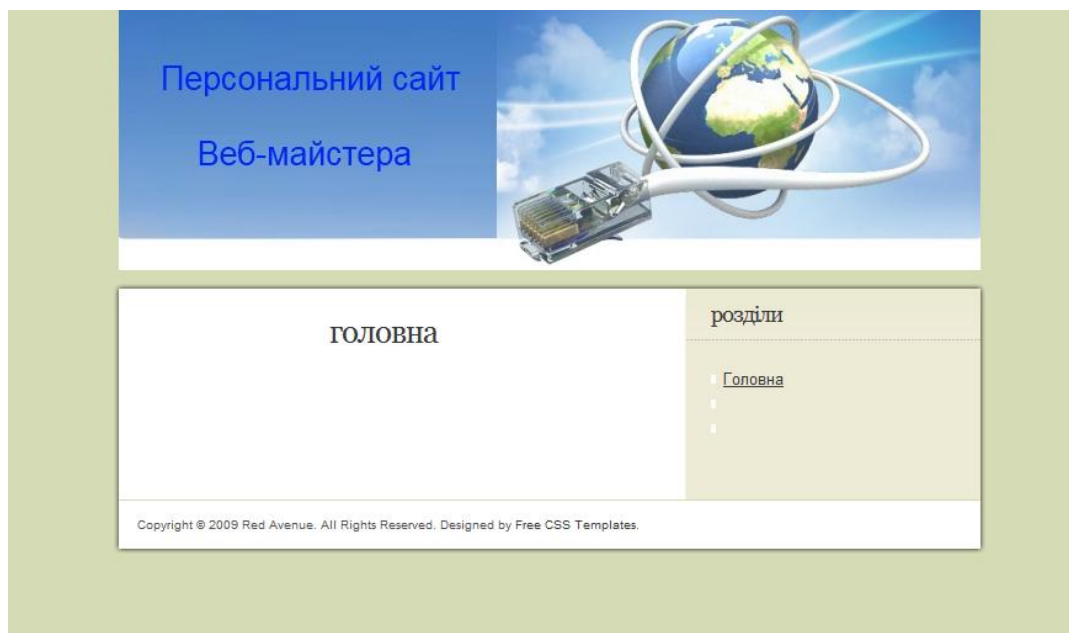


Рисунок 3.20 - Головна сторінка сайту.

Після цього починаємо заповнювати наш сайт контентом і необхідними сторінками (рисунок 3.21). Головна сторінка створюється автоматично без заголовку і тексту. Заходимо в пункт навігації “всі сторінки” і редагуємо головну сторінку веб-ресурса (рисунок 3.22).

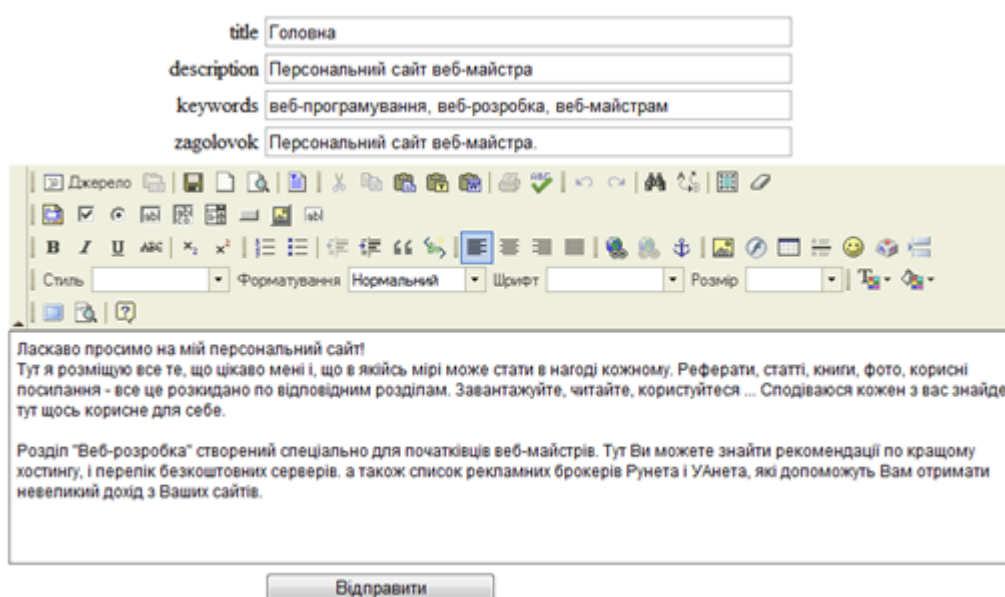


Рисунок 3.21 - Сторінка заповнення категорії контентом

										Арк.
										72
Змн.	Арк.	№ докум.	Підпис	Дата						



Рисунок 3.22 - Головна сторінка веб-ресурса

Добавляємо ще один розділ “Веб-програмування” і до нього ми вже добавимо підрозділи сайту. Знову заходимо дадати сторінку, вводимо дані сторінки і в нижньому меню вибираємо `tormenu` (рисунок 3.23).

Назва файла

Добавити в меню: ▼

Рисунок 3.23 - Вибір категорії меню

Потім заходимо на сторінку редагування меню в таблиці “Добавляємо сторінки меню”, вибираємо сторінку (рисунок 3.24), назву і розділ, в якому дана сторінка буде розміщена (рисунок 3.25).

Добавляємо сторінки меню

Сторінка	Назва	Розділ
index		ні

Добавити

Рисунок 3.24 - Добавлення сторінки



Рисунок 3.25- Створена сторінка веб-ресурса

Створюємо голосування на сайті, заходимо в пункт навігації «створити голосування». В полі для відповідей потрібно вносити кожен відповідь з нового рядка, як показано на рисунках 3.26 та 3.27.

адмін панель

Настроїти | Редагувати | Нове голосування

Створення голосування
В полі для відповідей потрібно вносити кожен відповідь з нового рядка.

Запитання: Якій мові програмування ви віддаєте перевагу

Відповіді:

- PHP
- Perl
- C#
- JavaScript
- Vbscript
- інші

Створити

2010 WEB-if

Рисунок 3.26 - Сторінка створення голосування



Рисунок 3.27- Головна сторінка веб-ресурса.

3.4 Захист програмного сервісу

Аутентифікація - процес, за допомогою якого перевіряється, що хтось є саме тим, за кого він себе видає. Як правило, перевірка включає в себе введення імені та пароля. Розглянемо, як працює базова аутентифікація. При зверненні відвідувача у захищену директорію, сервер Apache у відповідь на запит посилає заголовок з кодом 401 (401 authentication required header). Браузер відвідувача приймає заголовок з кодом 401 і виводить вікно з полями для введення імені користувача і пароля. Після введення імені та пароля ці дані відсилаються назад на сервер, який перевіряє ім'я користувача в спеціальному списку, а пароль на правильність. Якщо все вірно, то відвідувач отримує доступ до ресурсу. Разом із заголовком браузеру надсилає спеціальне ім'я, зване областю дії. Браузер кешує не лише ім'я, але і пароль, щоб передавати їх при кожному запиті.

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			75

Завдяки цьому введення імені та пароля в захищану директорію здійснюється тільки раз. В іншому разі їх необхідно було б вводити при кожному запиті до захищеної директорії. Кешування параметрів аутентифікації (ім'я, пароль, область дії), зазвичай здійснює тільки в межах одного сеансу.

Але при базовій аутентифікації ім'я користувача і його пароль передаються в мережу у відкритому вигляді на протязі всього сеансу, коли відвідувач працює з захищеної директорією. Хакер може перехопити цю інформацію, використовуючи мережевий аналізатор пакетів.

Для того щоб захистити сайт, потрібно виконати наступну послідовність дій: створити файл з паролями, переписати його на сервер, створити файл. Ntaccess і теж переписати його на сервер. Для організації захисту знадобиться:

- 1) WEB-сайт і FTP-доступ до нього;
- 2) права на створення файлів Ntaccess і організацію захисту за допомогою них;
- 3) утиліта генерації паролів htpasswd.exe

3.4.1 Перевірка роботи файлу Ntaccess на сервері

Для того, щоб перевірити, чи є у організації права на захист за допомогою файлів Ntaccess створюється текстовий файл з ім'ям .Ntaccess (першим символом йде крапка, розширення відсутнє).

Перед тим як зберігається файл потрібно вписати рядки:

AuthType Basic

AuthName admin

require valid-user

Потім, за допомогою FTP-доступу, переписати файл .Ntaccess на веб-ресурс, в директорію, адмінпанелі.

Після цього відкривається запит на введення логіна і пароля. Файл з паролями створюється утилітою htpasswd.exe. Дана утиліта знаходиться в теці з встановленим Apache у підкаталозі bin.

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			76

Для роботи з утилітою `htpasswd.exe` необхідний інтерфейс роботи з командним рядком. Інтерфейсом роботи з командним рядком володіють такі програми, як `Far`, `Windows Commander` і т.п.

Натискаємо "Пуск" -> "Виконати" і відкриється вікно утиліти.

Далі необхідно перейти в директорію, де знаходиться утиліта `htpasswd.exe`. введемо у командний рядок команду `usr/local/apache/bin` (це шлях де знаходиться утиліта). Після цього потрібно дати команду на створення файлу з паролем.

Введемо у командний рядок наступне:

- `htpasswd-cm. htpasswd admin;`
- *-`Cm` - це ключі для утиліти. Ключ `c` - вказує, що необхідно створити новий файл з паролями. Якщо файл з таким ім'ям вже існує, то він буде перезаписаний. Ключ `m` - визначає шифрування за алгоритмом MD5;
- *. `Htpasswd` - ім'я файлу з паролями (можна використовувати будь-яке ім'я);
- * `Admin` - ім'я відвідувача, якому буде дозволено доступ до захищеної частини.

У відповідь повинен з'явитися відповідний запит на введення пароля і його повторення. Якщо все правильно, то в завершенні має з'явитися повідомлення: `Adding password for user admin`. І в директорії `c: Apache in` з'явиться файл `.Htpasswd`, в якому буде знаходитися рядок з ім'ям користувача і хеш-кодом його пароля. Для того, що б в той же файл `.Htpasswd` додати ще одного користувача, необхідно прибрати ключ з команди запуску утиліти `htpasswd.exe`:

`htpasswd-m. htpasswd admin`

Отже, файл з паролями створений. Тепер необхідно переписати його на сервер. Файли з паролями дуже бажано класти вище кореневої директорії - туди, куди не буде доступу відвідувачам.

									Арк.
									77
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

3.4.2 Створення файлу .htaccess

Для захисту директорії можуть використовуватися наступні директиви:

- * AuthType - тип використовуваної аутентифікації. Для базової аутентифікації ця директива повинна мати значення: Basic;
- * AuthName - ім'я області дії аутентифікації. Текст, що допомагає відвідувачеві зрозуміти, куди він намагається отримати доступ. Наприклад, може бути написано: "Private zone. Only for administrator!";
- * AuthUserFile - шлях до файлу з паролями (.htpasswd);
- * AuthGroupFile - шлях до файлу груп, якщо він існує;
- * Require - одне або декілька вимог, які повинні бути виконані для отримання доступу до закритої області.

Приклад файлу .htaccess:

```
AuthType Basic
```

```
AuthName "Private zone. Only for administrator!"
```

```
AuthGroupFile /usr/host/mysite/group
```

```
AuthUserFile /usr/host/mysite/.htpasswd
```

```
require group admins
```

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

4 ОХОРОНА ПРАЦІ

Основною метою розділу охорони праці є уникнення можливості виробничого травматизму, професійних отруєнь і захворювань, пожеж і вибухів, аварій, забруднення довкілля при будівництві та використанні об'єкта проектування [16].

Даний дипломний проект передбачає розробку РНР-орієнтованої системи інтерактивного керування веб-ресурсами.

В розділі охорона праці проводиться розрахунок безпечних умов праці для приміщення з комп'ютерами [17].

Обслуговування апаратури виконується в кімнаті розміщення обладнання контролера базових станцій. Обслуговуючий персонал займається контролюванням роботи апаратури, виявленням аварій та їх усуненням.

Контроль за роботою здійснюється за допомогою комп'ютерного обладнання, тому ця робота відноситься до категорії легких, які виконуються в сидячому, стоячому положенні або пов'язані з незначним рухом, але вона не відноситься до систематичної фізичної роботи або до перенесення важких предметів.

Виходячи зі СН 245-71 і ГОСТ 12.1.005-88, а також, беручи до уваги характер робіт, відповідно до яких, площа приміщення на одного працівника в приміщенні дорівнює (6 м^2), приймаємо:

$$S_n = n S_0, \quad (4.1)$$

де S_0 – площа приміщення, що виводиться на одного працівника;

n – кількість працівників.

Оскільки в приміщенні працює 6 чоловік, тоді необхідна площа для роботи повинна становити: $S_n = 6 \cdot 6 = 36 \text{ м}^2$.

Реальна площа приміщення становить 48 м^2 , тобто відповідає вимогам санітарних норм.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

Згідно ГОСТ 12.1.005-88, в приміщенні повинні підтримуватися певні метеорологічні умови, що визначаються температурою відносною вологістю повітря, тиском та швидкістю руху повітря. Ці фактори впливають на терморегуляцію, тобто спроможністю організму людини підтримувати нормальну температуру тіла (в межах 36 – 37 °С).

Тепловіддача від організму може здійснюватись шляхом тепловипромінення, конвекції і випаровування. При підвищеній температурі навколишнього середовища тепловіддача здійснюється лише за рахунок випаровування поту. Перегрівання тіла до 40 – 41 °С приводить до порушення водно – сольового обміну, виникнення судорожної хвороби і теплового удару з втратою свідомості.

Робота в умовах пониженої температури повітря, особливо при підвищеній вологості і швидкості руху, призводить до переохолодження тіла, що супроводжується виникненням простудних захворювань. Мінусова температура повітря призводить до обморожування, що розглядається як виробнича травма.

Для робочої зони нашого приміщення оптимальні і допустимі значення температури, відносної вологості і швидкості руху повітря встановлюються з врахуванням трудоемності і складності роботи, яка виконується. Користувачі персональних комп'ютерів належать до групи 1а – легкі роботи.

Відповідно з цим і ГОСТ 12.1.005–88 вибираємо необхідні метеорологічні умови (таблиця 4.1).

Для підтримання відповідних метеорологічних умов в приміщенні встановлено обладнання системи центрального опалення, але в зимовий період його тепловіддача є недостатньою. Доцільним є проведення ущільнення конструктивів вікон і дверей, щоб припинити втрати тепла.

Решту метеорологічних умов забезпечує обладнання повного кондиціонування повітря. Воно забезпечує постійність температури, вологості, руху і чистоти повітря.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 – Оптимальні та допустимі метеоумови

Період року	Категорія робіт	Температура, °С		Відносна вологість повітря		Швидкість повітря	
		оптимальна	допустима	оптимальна	допустима	оптимальна	допустима
Холодний	Легка 1а	22–24	21–25	40–60	35–75	0,1	0,05– 0,2
Теплий	Легка 1а	23–25	22–28	40–60	35–75	0,1	0,05– 0,2

Сприятливі умови роботи забезпечують як високу продуктивність праці, так і позитивно впливають на психологічний стан людини, на її працездатність і здоров'я. Особливо важливе біологічне і гігієнічне значення для людини має природне освітлення, тому при проектуванні виробничих приміщень важливо передбачити наявність природного освітлення СНиП II–4–79.

Проведемо розрахунок природного освітлення згідно зі СНиП II–4–79 «Природне і штучне освітлення. Норми проектування», а при необхідності розрахуємо додаткове штучне освітлення приміщення.

Розрізняють три системи природного освітлення: бокове, верхнє, комбіноване. Для кількісної оцінки виробничого освітлення важливою технічною характеристикою є освітленість робочої поверхні. Густина світлової енергії на площі E (лк) визначається за формулою:

$$E = dF/dS, \quad (4.2)$$

де dF – світловий потік, який характеризує потужність світлового випромінювача (лм), рівномірно розподілений по площі dS (м²).

Коефіцієнт природного освітлення, який являє собою відношення освітленості в даній точці середини приміщення $E_в$ до зовнішнього горизонтального освітлення $E_з$ визначаємо за формулою:

$$I = E_в / E_з. \quad (4.3)$$

Заміри натурального освітлення проводяться люксометром 10116.

Розміри приміщення становлять:

$E_n \cdot B = 6 \times 8 \text{ м}^2$; висота приміщення $h = 3 \text{ м}$, S – світловий отвір вікон $1 - 1,9 \text{ м}^2$.

Віконне скло подвійне. Характеристика зорової роботи відноситься до високої точності. Це відповідає нормі природного освітлення КПО $I_n = 2 \%$ при боковому освітленні.

При боковому освітленні використовується формула:

$$100 \frac{S_0}{S_n} = \frac{I_n \cdot K_з \cdot \eta_{10}}{\tau_0 \cdot VI} K_б; \quad (4.4)$$

де S_0 – площа світлових отворів, м^2 ;

S_n – площа підлоги, м^2 ;

$K_з$ – коефіцієнт світлопроникнення;

η_{10} – світлова характеристика вікон;

τ_0 – загальний коефіцієнт світлопроникності;

VI – коефіцієнт, який враховує відбивання світла від поверхні;

$K_б$ – коефіцієнт, який враховує затемнення будинками, що стоять навпроти.

Для приміщення розмірами $6 \cdot 8 \cdot 3$ площа $S = 48 \text{ м}^2$; для $L_n/B = 8/6 = 1,33$;
 $B/H = 6/3 = 2$; $\eta_{10} = 16$.

Для середньозваженого коефіцієнта відображення стелі, стін і підлоги, який дорівнює 0,4, коефіцієнт VI становить 2,4, $K_б$ приймаємо – 1,4.

Для приміщень з повітряним середовищем, в якому концентрація пилу менше 1 мг/м^3 $K_з = 1,4$; оскільки $I_n = 2 \%$, коефіцієнт τ_0 визначаємо за формулою:

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \cdot \tau_5; \quad (4.5)$$

де τ_1, τ_2, τ_3 – коефіцієнти світлопропускання матеріалу вікна, виду вікна і його конструкції: для віконного, листового, подвійного скла $\tau_1=0,8$; для дерев'яних подвійних роздільних оправ до вікон $\tau_2=0,6$; для залізобетонних конструкцій $\tau_3=0,8$;

τ_4 – коефіцієнт, який враховує витрати світла в сонцезахисних конструкціях: для жалюзі і штор, що регулюються, дорівнює 1;

τ_5 – коефіцієнт, який враховує втрати світла в захисній сітці, що встановлюється під світильником — дорівнює 0,9.

Отже: $\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1 \cdot 0,9 = 0,35$.

Визначаємо площу світлових отворів S_0 :

$$S_0 = \frac{I_n \cdot K_z \cdot \eta_{10} \cdot S_n}{100 \cdot \tau_0} = K_5; \quad (4.6)$$

де S_n — стандартна площа вікна.

Кількість вікон визначаємо за формулою:

$$S_0 = \frac{2 \cdot 1,4 \cdot 16 \cdot 1,4 \cdot 48}{100 \cdot 0,35 \cdot 24} = 3,47 (\text{м}^2) \quad (4.7)$$

Відповідно: $n = 3,47/1,9 = 1,83 = 2$ вікна. Таким чином, для забезпечення КПО $I_n = 2\%$ у приміщенні повинно бути два вікна площею $1,9 \text{ м}^2$.

Для освітлення приміщення, коли природного освітлення недостатньо або взагалі немає, використовується штучне освітлення.

Світловий потік Φ – це потужність світлової енергії, що оцінюється за світловим відчуттям, яке воно справляє на органи зору людини:

$$\Phi = dQ/dt. \quad (4.8)$$

									Арк.
									83
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

Сила світла I – це відношення світлового потоку до величини тілесного кута, в якому рівномірно розподілено випромінювання:

$$I = dF/d\omega. \quad (4.9)$$

Освітленість E – густина світлового потоку на освітлюваній поверхні:

$$E = d\Phi/dS. \quad (4.10)$$

Яскравість L – поверхнева густина сили світла у заданому напрямку:

$$L = dl/dS \cdot \cos(\alpha). \quad (4.11)$$

Коефіцієнт відбиття β – відношення відбитого світлового потоку до падаючого: $\beta = \Phi_{\text{відб}}/\Phi_{\text{пад}}$.

Фон – поверхня, що прилягає безпосередньо до об'єкта розпізнавання, на який цей об'єкт сприймається. Фон характеризує коефіцієнт відбиття (залежить від кольору поверхні та від її фактури). Фон світлий $\Phi > 0,4$; середній – $\Phi = 0,2 - 0,4$; темний $\Phi < 0,2$.

Контраст – ступінь розпізнавання яскравості об'єкта і фону:

$$K = (L_0 - L_{\phi}) / L_0. \quad (4.12)$$

Контраст великий, то $K > 0,5$; середній – $K = 0,2 - 0,5$; маленький – $K < 0,2$.

Коефіцієнт пульсацій K_n – критерій оцінки відносної глибини коливань освітленості в результаті зміни в часі світлового потоку газорозрядних ламп при живленні їх змінним струмом:

$$K_n = (E_{\text{макс}} - E_{\text{мін}}) \cdot 100\% / (2 \cdot E_{\text{сер}}) \quad (4.13)$$

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

де $E_{сер}$ – значення освітленості за період.

Розміри приміщення: $A = 8$ м, $B = 6$ м, $H = 3$ м. Нормована освітленість 300 лк. Показник приміщення: $i = A \cdot B / (H \cdot (A + B)) = 8 \cdot 6 / (3 \cdot (8 + 6)) = 1,14$.

Вибираємо світильник НОДЛ з коефіцієнтом використання світлового потоку $\eta = 49\%$. Сумарний світловий потік:

$$\Phi = ((E_n \cdot S \cdot k \cdot Z) / \eta) \cdot 100\%, \quad (4.14)$$

де E_n – нормована освітленість, лк;

S – площа приміщення, м²;

k – коефіцієнт запасу;

Z – коефіцієнт мінімальної освітленості;

η – коефіцієнт використання світлового потоку.

$$\Phi = ((300 \cdot 48 \cdot 1,75 \cdot 1,1) / 49) \cdot 100\% = 56\,572 \text{ лм.}$$

Вибираємо лампи ЛТБ-80 р, $\Phi_{л}$ — 4300 лм, тоді кількість ламп дорівнює: $N = \Phi / \Phi_{л} = 56572 / 4300 = 14$ шт. Кількість світильників: $N_c = N / 2 = 7$ шт.

Перерахуємо значення E :

$$E = \frac{N \cdot \Phi_{л} \cdot \eta}{S \cdot k \cdot Z \cdot 100\%} = \frac{14 \cdot 4300 \cdot 49}{48 \cdot 1,75 \cdot 1,1 \cdot 100\%} = 319,3. \quad (4.15)$$

Отже, штучне освітлення забезпечує освітленість $E = 319$ лк, що є більшим за $E_n / E_n = 300$ лк, тобто розрахунок проведений правильно.

Рівень шуму дорівнює 75 дБ, що відповідає вимогам ГОСТу, тому захисних заходів не передбачається.

Електричний струм при дії на людину може викликати як місцеві, так і загальні пошкодження. Місцеві електротравми – це опіки, нагрівання внутрішніх органів, механічні пошкодження (розрив тканин м'язів), порушення біоелектричних процесів у організмі, електроліз органічних рідин. Зовнішніми проявами електротравм можуть бути термічні опіки, електричні ознаки на шкірі, металізація

									Арк.
									85
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ				

поверхні шкіри, електроофтальмія (ураження зору під дією ультрафіолетових променів при іскровому розряді). Загальне ураження струмом відбувається при проходженні струму через нервові центри, центри дихання і роботи серця (електричний удар).

Небезпека ураження тим більша, чим більший струм проходить через людину, але крім цього, впливають: тривалість і шлях проходження струму, його вид, частота і виробничі умови.

Умови ураження людини електричним струмом такі:

- двофазне дотикання (двофазне включення людини в мережу);
- однофазне дотикання, наближення на небезпечну віддаль до неізольованих дротів з напругою більше 1000 В;
- дотик до корпусу обладнання, що не проводить струм, але опинилося під напругою;
- перебування в зоні дії атмосферної електрики;
- вхід у зону дії електромагнітного поля.

Згідно класифікації приміщень за ступенем небезпеки ураження електричним струмом (ПУЕ 1.1.6) приміщення роботи системи відноситься до першого (без підвищеної небезпеки).

Електричні установки, до яких відноситься переважна більшість обладнання системи, вимагають дотримання правил електробезпеки, оскільки в процесі експлуатації або проведення профілактичних робіт людина може доторкнутись до частин, що знаходяться під напругою 220 В, тому виникає необхідність у захисті персоналу від ураження електричним струмом. Дуже велике значення для запобігання електротравматизму має правильна організація експлуатації, обслуговування системи. Під цим розуміється точне виконання ряду організаційних та технічних заходів, які встановлені діючими «Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів» (ППЕ і ПТБ споживачів) і «Правилами побудови електропристроїв» (ППЕ). Основними технічними засобами, які забезпечують безпеку робіт в електроустановках, є: захисне

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			86

заземлення, занулення, вирівнювання потенціалів, захисне включення, електричний розподіл мереж, мала напруга, подвійна ізоляція. Використання цих засобів у різноманітних поєднаннях дозволяє захистити людину від ураження струмом.

Захисне заземлення – це навмисне електричне з'єднання з землею або її еквівалентом металевих неструмопровідних частин, які можуть бути під напругою. У приміщенні розміщення контролера базових станцій заземлено всі шафи з обладнанням, а також вся комп'ютерна техніка. Приміщення, де знаходиться система, обладнується контуром-шиною захисного заземлення, яка з'єднується із заземлювачем. Контур-шина виготовляється з мідного дроту діаметром 6 мм у перерізі і вкладається по периметру приміщення. Місця перетину дротів пропаюються з застосуванням бікислотного флюсу. Для під'єднання заземлювальних провідників на шину наварюються гвинти М8. У дипломній роботі проведу розрахунок захисного заземлення згідно порядку, встановленого ПУЕ.

Згідно вимог ПУЕ 1.7.65 в електроустановках з напругами до 1 кВ при потужності трансформатора менше 100 кВт опір заземлювача повинен бути не більше 10 Ом.

1. Визначаємо розрахунковий опір землі: $r_{op.z.} = \Phi r_{o3}$, де Φ – коефіцієнт сезонності, який враховує коливання питомого опору при зміні вологості ґрунту протягом року; використовується стержневий заземлювач (рисунок 4.1) довжиною $l = 2$ м при глибині закладання від вершини $h = 0,5$ м, $\Phi = 1,1$ для четвертої кліматичної зони. Питомий опір ґрунту: $r_{o3} = 300$ Ом·м - для піску; $r_{op.z.} = 1,1 \cdot 300 = 330$ Ом·м.

2. Визначаємо опір R , розтікання струму в землі від одного вертикального заземлювача:

$$R_B = \frac{r_{on.z.}}{2 \cdot 3,14 \cdot l} \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.16)$$

						ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			87

де l – довжина заземлювача ($l = 2$ м);

$d=0,05$ м – діаметр заземлювача за таблицею при $U < 1$ кВ та при $S < 100$ кВА;

t – відстань від поверхні землі до середини заземлювача,

$$t = h + l/2 = 0,5 + 2/2 = 1,5 \text{ м}; R_B = \frac{330}{2 \cdot 3,14 \cdot 2} \left(\ln \frac{2 \cdot 2}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 1,5 + 2}{4 \cdot 1,5 - 2} \right) = 133,3 \text{ Ом.}$$

3. Приблизна кількість заземлювачів: $n = \frac{R_6}{R_{6,ннмм}} = \frac{133,3}{10} = 13,3 \approx 14$.

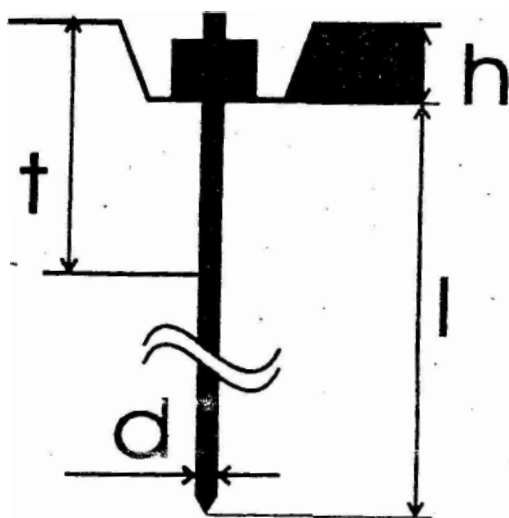


Рисунок 4.1 – Схема розташування одного заземлювача в ґрунті

4. Знаходимо із таблиць коефіцієнт використання вертикальних заземлювачів, який враховує ефект екранування при вибраному значенні $k = a/l$, де a — віддаль між заземлювачами, м; $k = 1,2$ при $a = 2,4$ м; отже коефіцієнт використання вертикального заземлювача за таблицями дорівнює $\eta_e = 0,56$.

5. Кількість вертикальних заземлювачів з урахуванням η_e обчислюємо за формулою $n = \frac{R_6}{R_{6,норм} \cdot \eta_e} = \frac{133,277}{10 \cdot 0,56} = 23,799 = 24$.

6. Довжина горизонтального заземлювача для розміщення по контуру $L = a \cdot n = 2,4 \text{ м} \cdot 24 = 57,6 \text{ м}$.

7. Опір горизонтального заземлювача R_r (Ом), прокладеного на глибині $h = 0,5$ м від поверхні землі:

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

$$R_z = \frac{r_{o.p.k.}}{2 \cdot 3,14 \cdot L} \ln \frac{2 \cdot L}{b \cdot h} = \frac{330}{2 \cdot 3,14 \cdot 57,6} \ln \frac{2 \cdot 57,6}{0,04 \cdot 0,5} = 7,3 \quad (4.17)$$

де $b = 0,04$ м — ширина штабової сталі, з якої виготовлений заземлювач.

8. Обчислюємо загальний опір:

$$R_k = \frac{R_g \cdot R_o}{n \cdot R_o \eta_g + R_g \eta_d} = \frac{133,3 \cdot 7,3}{24 \cdot 7,3 \cdot 0,56 + 133,3 \cdot 0,27} = 7,5 \text{ Ом} \quad (4.18)$$

Результат є менше 10 Ом, тобто виконується нормуюча умова $R_z < R_{z,норм.}$.

Велика увага приділяється дотриманню обслуговуючим персоналом правил роботи в приміщенні, яке призначене для експлуатації системи. У приміщенні не повинно бути сторонніх людей. Працівники повинні використовувати спецодяг. Безпека роботи обслуговуючого персоналу в приміщенні забезпечується:

- наявністю нормальних проходів між обладнанням;
- використанням спеціальних технічних меблів;
- використанням електрозахисних засобів (діелектричних килимків, гумових рукавиць);
- наявністю аварійного освітлення ($E=2$ лк);
- обладнанням розеток з напругою 220 В;
- заземленням корпусів обладнання і апаратури освітлювальних пристроїв.

Одне з основних місць в охороні праці займає пожежна безпека.

Першочергове завдання пожежної профілактики — це запобігання пожеж. Під пожежною профілактикою розуміють комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожеж, обмеження їх розповсюдження, а також на створення умов для успішного гасіння пожеж. Пожежно-профілактичні заходи розробляються

											Арк.
											89
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КСМ.19110/11.00.00.000.ПЗ						

та виконуються разом, в тісному взаємозв'язку з усіма проектними, будівельними та експлуатаційними роботами.

Приміщення чергування технічного персоналу забезпечується протипожежним інвентарем (вуглекислотними вогнегасниками типу ВВ-2). Проходи між рядами і вихід не повинні загроможуватись. У випадку виникнення пожежі перш за все потрібно виключити джерело живлення, сповістити про пожежу в пожежну частину. Евакуювати сторонніх людей, які могли опинитися в небезпечній зоні і лише після цього приступити до гасіння пожежі і рятування цінного обладнання.

Один вуглекислотний вогнегасник ВВ-2 розрахований на 40–50 м² приміщення. Для ліквідації невеликих пожеж можна використовувати деякі порошкові матеріали (хлориди лужних металів, соду, пісок і т. д.), що подаються в зону горіння порошковими вогнегасниками.

Будівля, в якій знаходиться наше приміщення, обов'язково має резервний вихід на випадок екстреної евакуації працівників і неможливості використання основного виходу.

За вибухопожежною і пожежною безпекою приміщення і будівлі згідно ОНТП-24-86 і СНТП 2.09, СНТП 02-85 діляться на категорії А, Б, В, Г, Д.

Для приміщення чергування персоналу встановлена категорія пожежної безпеки Д (СНІП 2.09.02-85) при ступені вогнестійкості (СНІП Н-90-81), що означає наявність у приміщенні негорючих речовин та матеріалів у холодному стані.

Для швидкого сповіщення пожежної сходи при виникненні пожежі приміщенні використовується електрична пожежна сигналізація. Система електричної пожежної сигналізації виявляє пожежу на початковій стадії і сповіщає про місце її виникнення, а також автоматично включає стаціонарні установки гасіння пожеж.

Автоматичні сповіщувачі при ознаках пожежі здійснюють посилення сигналу. Сповіщувачі типу АТИП-1, АТИП-3 і АТИП-3М спрацьовують внаслідок теплової деформації (при 80–100 °С) біметалічних пластинок і мають

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

розраховану площу обслуговування в приміщеннях до 15 м². Комбіновані теплові і димові сповіщувачі типу КИ-1 мають чутливий елемент у вигляді іонізуючої камери (реагування на дим) і терморезистори (реагування на тепло). Температура спрацювання цих сповіщувачів 50–80°С, площа обслуговування 100 м².

Передбачені нами заходи з охорони праці в першу чергу призначені для уникнення нещасних випадків, що можуть виникнути на підприємстві.

В іншому передбачені заходи з охорони праці відповідають вимогам нормативних документів та актів та забезпечують нормальну, ефективну і безпечну для здоров'я людини виробничу діяльність

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломному проекті було проаналізовано принципи розробки систем інтерактивного керування веб-ресурсами і їх недоліки, на основі цього був розроблений програмний сервіс.

Досліджено структуру та функціональність систем. Проаналізовано подібні програмні комплекси та на основі їх переваг і недоліків сформовано постановку задачі. Було проведено розробку моделі та схеми вимірювання якості програмних ресурсів. Розроблено структуру програмного сервісу. Здійснено опис основних алгоритмів і процедур програмного сервісу.

Розроблений програмний сервіс дозволяє істотно знизити навантаження для створення і керування веб-ресурсами простому користувачеві який не має спеціальних навичок і вмінь. А також розроблений інтерфейс, заснований на описі задач.

Програмний сервіс є гнучким та універсальним, щоб подальшому можна розширювати його функціональність.

В останньому розділі був проведений аналіз питань охорони праці при впровадженні розробленої системи звукового забезпечення (Додаток Д).

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пелецишин А.М. Методи визначення та оптимізації тематики сайту // Вісник НУ "Львівська Політехніка" Інформаційні системи та мережі. – 2004. – №519. – С.254–267.
2. Пелецишин А.М., Гулка Т.Б. Інформаційна система аналізу діяльності Web-вузла // Вісник НУ "Львівська Політехніка" Інформаційні системи та мережі. – 2001. – №438. – С.115–120.
3. Carroll J., Rosson M. Better Home Shopping or New Democracy? Evaluating Community Network Outcomes // Human-Computer Interaction and Computer Science. – 2001. – №5. – P.8–12.
4. Буров Є. Комп'ютерні мережі. – Львів: Бак, 1999. – 468 с.
5. Гук М. Аппаратные средства IBM PC. – Спб.: Питер, 2006. – 1072 с.
6. Дейт К. Дж. Введение в системы баз данных: Пер. с англ. – 6-е изд. – К.: Диалектика, 2007. – 784 с.
7. Ситник В.Ф. Системи підтримки прийняття рішень. – К.: Техніка, 2005. – 164с.
8. Хоумер А., Улмен К. Dynamic HTML: справочник. – СПб.: Питер, 2000. – 465 с.
9. Аткинсон Л., Сураски З. PHP5: Библиотека профессионала, 3-е издание: Пер. з англ. – М.: Вильямс, 2006. – 944 с.
10. Converse T., Park J., Morgan C. PHP5 and MySQL Bible. – Indianapolis, Canada: Wiley Publishing Inc., 2004. – 1083 p.
11. Harris A. PHP/MySQL Programming for the Absolute Beginner. – London: Premier Press, 2003. – 785 p.
12. Томпсон Л., Веллинг Л. Разработка WEB-приложений с помощью PHP и MySQL, 2-е издание.: – М.: Вильямс, 2006. – 894 с.
13. Митчелл М., Оулдем Д., Самьюэл А. Программирование для Linux. Профессиональный поход. – М.: Вильямс, 2002. – 288 с.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		93

- 14.Лапінський В. В., Габрусев В. Ю. Основи операційних систем: Посібник для студентів. – К.: Вища школа, 2007. – 96 с.
- 15.Таненбаум Э., Вудхалл А. Операционные системы: разработка и реализация. Классика CS. – СПб.: Питер, 2006. – 576 с.
- 16.Ткачук К.Н. Справочник по охране труда. – К.: Техніка, 1991. – 110с.
- 17.Методичні вказівки до написання розділу «Охорона праці» в дипломних проектах з освітньо–кваліфікаційного рівня «Спеціаліст» для спеціальності 7.091501 «Комп’ютерні системи та мережі» / Г.В.Сапожник, Н.М.Васильків. – Тернопіль: ТАНГ, 2004. – 24 с.
- 18.Методичні рекомендації до виконання дипломного проекту з освітньо–кваліфікаційного рівня „Спеціаліст”. Спеціальність „Комп’ютерні системи та мережі” / О.М.Березький, Н.М.Васильків, І.В.Васильцов, Р.Б.Трембач / Під ред. М.П.Карпінського – Тернопіль : ТНЕУ, 2008. – 38 с.

					ДП.КСМ.19110/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		94