

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри

_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Веб система електронної біржі праці

Освітньо-кваліфікаційний рівень “бакалавр”
Спеціальність 6.050102 – “комп’ютерна інженерія”

Керівник проекту:

(підпис)

Кривуля Г.Ф.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Відчинкін О.А.

(ініціали, прізвище)

Група:

КІ-14з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний
рівень бакалавр
Напрямок підготовки 6.050102 – “комп'ютерна інженерія”
(шифр і назва)
Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ
І.С. Скарга-Бандурова
« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Відчинкіну Олександрю Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Веб система електронної біржі праці

керівник проекту (роботи) Кривуля Геннадій Федорович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " _____ " _____ 201_ р. № _____

2. Термін подання студентом роботи 16.06.2018

3. Вихідні дані до роботи Теоретичні відомості про особливості
працевлаштування, теоретичні відомості про особливості бізнес процесу,
теоретичні відомості про Web інтерфейс системи, середа розробки веб
платформи PHPStorm8

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) Аналіз предметної області, архітектура системи, програмна
реалізація та її аналіз, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2018

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз завдання та робота з літературою	05.05.2018 - 13.05.2018	
2	Розробка веб платформи	14.05.2018 - 22.05.2018	
3	Тестування веб платформи	22.05.2018 - 02.06.2018	
4	Розробка розділу «Охорона праці»	02.06 .2018- 11.06.2018	
5	Оформлення пояснювальної записки та електронних плакатів	11.06.2018 - 16.06.2018	

Здобувач вищої освіти

_____ (підпис)

Відчинкін О.А.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Кривуля Г.Ф.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка випускної кваліфікаційної роботи бакалавра: 79с., 16 рис., 1 табл., 27 джерел, 2 додатки.

Метою дипломної роботи бакалавра є розробка та подальше впровадження веб системи електронної комерції з працевлаштування та ведення бізнесу. Розроблена Web система котра забезпечує знаходження потрібної вакансії для здобувачів або роботодавців. Правила дозволяють визначити який більш доцільний варіант вакансії чи резюме є для обраного користувача, виходячи з її характеристик. Система заснована на фреймворку Laravel що сприяє стандартам розробки PSR-4 та робить систему більш гнучкою та захищеною. Вхідними даними для системи є база даних та мануальне введення.

Веб система створена у середовищі розробки PHPStorm актуальної версії 8, з використанням вбудованих засобів побудови діаграм та допоміжних засобів розробки таких як робота з СУБД, SVN та FTP. Дана система підтверджує свою актуальність та ефективність.

Ключові слова: WEB СИСТЕМА, ЕЛЕКТРОННА КОМЕРЦІЯ, ЕЛЕКТРОННИЙ БІЗНЕС, HTML5, CSS3, PHPSTORM 8, LARAVEL

Умови одержання дипломного проекту: СНУ ім. В. Даля,
пр. Центральний 59-А,. м. Сєвєродонецьк, 93400.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 6

ВСТУП 7

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9

1.1 Відомості про Е-комерцію 9

1.2 Переваги та недоліки Е-бізнесу 10

1.3 Постановка задачі 11

2 АРХІТЕКТУРА СИСТЕМИ 15

2.1 Функціональна структура програми 15

2.2 Архітектура клієнт-сервер 17

2.3 Функціональні модулі системи 18

2.4 Основні функції системи 20

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЇЇ АНАЛІЗ 26

3.1 Застосовані мови програмування 26

3.2 Проектна База даних 27

3.3 Допоміжні технології Laravel 32

3.4 Допоміжні технології - JQuery 33

3.5 Опис програмної реалізації 33

3.6 Ролі та права користувачів 35

3.7 Зовнішній вигляд веб системи 36

4 ОХОРОНА ПРАЦІ 46

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників
проектованого об'єкту, що мають вплив на персонал 46

4.2 Заходи щодо техніки безпеки 48

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці 51

4.4 Рекомендації по пожежній безпеці 55

ВИСНОВКИ 59

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 60

ДОДАТОК А. ЛІСТИНГ КОДУ 63

ДОДАТОК Б. ЕЛЕКТРОННІ ПЛАКАТИ 75

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЕК - електронна комерція

Е-бізнес - електронний бізнес

FTP - File Transfer Protocol

PHP - Hypertext Preprocessor

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheets

СУБД - Система Управління Базами Даних

ВСТУП

На сьогоднішній день через Інтернет можна придбати практично будь-які товари та послуги.

Електронна комерція на сьогоднішній день це прискорення більшості бізнес-процесів за рахунок їх проведення електронним чином. У цьому випадку інформація передається прямо до одержувача, минаючи стадію створення паперової копії на кожному етапі. Таким чином, електронну комерцію можна характеризувати як ведення бізнесу через Інтернет. У сучасному суспільстві все більше компаній переносять значну частину ділового спілкування в Мережу.

Серед відмінних особливостей онлайн комерції перед традиційною торгівлею можна відзначити:

- відсутність географічних, часових і мовних бар'єрів, що дозволяє просувати послуги на нові ринки збуту;
- більш низький рівень витрат виробництва і обігу, що досягається шляхом впровадження нових технологій в усі сфери діяльності компаній: починаючи від закупівель сировини і матеріалів і закінчуючи дистрибуцією готової продукції та пост-продажним обслуговуванням;
- більш високий рівень конкуренції: відстань між магазинами всього кілька секунд - саме цей час необхідний для завантаження відповідного сайту.

Інтернет-комерція включає в себе інтернет-магазини, біржі та посередницькі інтернет-аукціони, операції взаємодії між підприємствами, організацію різних каталогів і засобів спілкування користувачів в Інтернеті, проведення рекламних кампаній певних товарів або ресурсів, і т.д. Число товарів і послуг постійно поповнюється, пропонуючи користувачам все більше число можливостей.

В останні роки одним з найбільш ефективних способів пошуку роботи стало використання сайтів із працевлаштування. Або, як кажуть рекрутери, «робітних» або «Джоб (job-) сайтів».

Проблема працевлаштування хвилює сьогодні багатьох. Найчастіше пошук роботи для здобувача - тривалий і болісний процес, який ускладнюється скороченням кількості вакансій і високим конкурсом на посади. Тим часом, нелегко доводиться і роботодавцям, які можуть витратити тижні, а то й місяці на пошук відповідного співробітника.

Однак сьогодні і здобувачі, і роботодавці можуть значно полегшити пошук придатних вакансій і резюме, контролюючи цей процес в будь-якому зручному для них місці за допомогою Інтернету. Тому особливу роль у відносинах "здобувач-роботодавець" грають портали з працевлаштування.

Метою розділу «Охорона праці» є створення безпечних і здорових умов праці на робочих місцях та в робочих зонах при розробці або експлуатації продукту проектування шляхом розробки питань охорони праці.

Основне завдання - виконати аналіз умов праці і характеру роботи програміста, виявити і усунути основні небезпечні та шкідливі виробничі фактори на робочому місці, провести оцінку впливу їх на організм людини.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Відомості про Е-комерцію

Е-комерція має безліч форм, у ній все частіше використовуються складні, інтегровані віртуальні торгові майданчики, що складаються з безлічі різних покупців і продавців.

Електронна комерція є операції, проведені за допомогою мережі Інтернет, в ході яких здійснюються покупки і продажі товарів і послуг.

Часто відбувається плутанина двох базових понять: е-комерції та е-бізнесу.

Згідно визначенню фахівців компанії ІВМ, електронний бізнес (е-бізнес) - перетворення основних бізнес-процесів при допомозі Інтернет технологій [1].

Таким чином, е-бізнесом ми називаємо будь-яку ділову активність, що використовує можливості глобальних інформаційних мереж для перетворення внутрішніх і зовнішніх зв'язків з метою створення прибутку.

Внутрішня організація компанії на базі єдиної інформаційної мережі (інтранет), що підвищує ефективність взаємодії співробітників і що оптимізує процеси планування і управління; зовнішня взаємодія (екстранет) з партнерами, постачальниками і клієнтами - все це складові частини е-бізнесу.

Електронна комерція є найважливішим складовим елементом е-бізнесу. Під електронною комерцією маються на увазі будь-які форми ділової операції, які проводяться за допомогою інформаційних мереж.

Виникнення і швидке розповсюдження в світі електронного бізнесу обумовлене поряд об'єктивно необхідних для цього умов і передумов. Перш за все, це високий рівень комп'ютеризації суспільства. Персональні комп'ютери, широкі, що отримали, розповсюдження у виробництві і побуті, стали матеріальною основою переходу більшої частини населення промислово розвинених країн до практичного оволодіння знанням і навиками

користування сучасними інформаційними технологіями. Крім того, для швидкого розповсюдження електронного бізнесу необхідно, щоб інформаційний і технологічний потенціал Інтернет перевищив деякий критичний рівень, що відкриває доступ до нього не тільки порівняно обмеженого круга елітарних користувачів, але і широких мас населення. Ще однією передумовою електронізації бізнесу є досягнення певного рівня інтеграції (інформаційною і технологічною) внутрішньо корпоративних інформаційних систем і Інтернет.

1.2 Переваги та недоліки Е-бізнесу

Електронні системи відрізняються від традиційних ринків низкою параметрів. По-перше, використання комп'ютерів автоматизує процес торгівлі та власне торгівельні відносини як між продавцями, так і між продавцями та клієнтами. По-друге, переваги використання ЕК на фінансових ринках полягають у зменшенні вартості, підвищенні ефективності та поліпшенні умов управління ризиками. Системи ЕК здійснюють безпосередню обробку даних, що дає змогу інтегрувати практично всі ланки бізнес процесу, починаючи від надання інформації про послугу і закінчуючи управлінням ризиками [2].

Таким чином, електронний бізнес сьогодні є:

- ефективним засобом проведення маркетингових досліджень та здійснення платежів;
- найкращою системою комунікацій, яка дає можливість установити та підтримувати постійний зв'язок з будь-яким абонентом у світі (за умов його підключення до мережі);
- потужним джерелом економічної, фінансової та наукової інформації.

Водночас повній реалізації потенціалу ЕК перешкоджають ряд чинників. Серед них можна умовно виділити такі:

- глобалізація - проблеми, пов'язані з мовним та культурним розмаїттям світової спільноти користувачів електронних мереж;
- договірно-правові та фінансові проблеми - ускладнення, пов'язані з укладанням угод через Інтернет, що виникають під час визначення часу, місця, юридичного статусу такої угоди, а також особливостей оподаткування;
- таємність та безпека - проблеми створення ефективних та надійних механізмів забезпечення конфіденційності, автентифікації (можливість для кожної із сторін угоди пересвідчитися в реальному існуванні одна одної) та цілковитого виконання всіх положень угоди. Більшість механізмів таємності та безпеки сьогодні передбачає сертифікацію третьою стороною (спеціалізованою установою або державою), що потребує створення глобальної сертифікаційної системи;
- взаємодія та сумісність систем - проблеми забезпечення універсальності стандартів взаємодії та сумісності при роботі в глобальній мережі незалежно від національних та географічних особливостей;
- залучення нових учасників - проблеми, пов'язані з недостатнім рівнем знань та кваліфікації окремих суб'єктів світової економіки (зокрема, середніх та малих підприємств), а також загальним технологічним відставанням окремих регіонів світу.

1.3 Постановка задачі

Необхідно організувати механізм реєстрації та авторизації користувачів, можливість додавання та вибірки з тематик каталогу, CRUT - операції, а так само підписку на поштову розсилку оголошень. Передбачити режим адміністрування з можливістю редагування основних параметрів програми та розділів каталогу оголошень, розділення користувачів з різними правами доступу.

Проведемо аналіз поставленого завдання: потрібно розробити веб-систему для організації електронної комерції з працевлаштування. Система працевлаштування призначена для невеликих і середніх комерційних організацій, охочих безкоштовно розмістити свої послуги в мережі інтернет, а так само приватних осіб.

Додаток повинен взаємодіяти з користувачами, тобто має бути інтерактивно, тому має бути написано на одній з мов програмування web-сценаріїв, таких як PHP або інших. Очевидно, що додатку потрібно оперувати з чималими масивами даних, тому для їх надійного зберігання потрібно база даних, оскільки випадок зберігання інформації безпосередньо в файлах на сервері не є надійним і безпечним через проблеми з множинним доступом. Додаток встановлюється на інтернет-сервері, доступ користувачів до додатка здійснюється через WEB-інтерфейс за допомогою браузера, наприклад MS Internet Explorer. Таким чином мається на увазі, що робота з додатком буде вестися тільки через HTTP - протокол.

Структура інтерфейсу електронної комерції з працевлаштування має бути зрозуміла для звичайного користувача, в той же час необхідно подбати про набір функціональних засобів, що забезпечують зручність роботи з набором оголошень. Для цього розділимо всі оголошення на рубрики і організуємо їх відображення у вигляді логічного дерева розділів і підрозділів каталогу. Оскільки у кожного оголошення є певний тип («попит», «пропозиція»), потрібно подбати про те, щоб користувач міг переглядати оголошення тільки потрібного йому типу.

При реєстрації у користувача запитується логін, пароль і контактна інформація. Для того щоб виключити зберігання паролів користувачів в базі даних у відкритому вигляді, потрібно передбачити їх шифрування у форматі MD5.

Зареєстровані користувачі проходять процедуру авторизації, в якій у них запитується логін і пароль. Так як використовується протокол HTTP, всі надіслані дані йдуть від користувача до сервера у відкритому вигляді. Для того

щоб звести до мінімуму ризик від перехоплення пароля, після процедури авторизації він ні в якому вигляді не повинен передаватися від сервера до користувача і навпаки. У теж час потрібно забезпечити подальшу аутентифікацію користувачів для визначення прав доступу при спробі виконанні ними певних операцій з даними. Отже, необхідно розробити такий алгоритм аутентифікації, при якому між сервером і користувачем передається лише деяка посилання-показчик на список успішно авторизованих користувачів. Після завершення роботи з додатком користувач посилає команду на видалення себе з цього списку, і посилання втрачає своє значення. У разі якщо користувач забув послати команду на видалення, посилання повинне віддалятися автоматично по закінченню певного проміжку часу. Бажано, щоб подібний показчик складався з випадкового набору великої кількості символів, тоді шанс перебрати всі посилання на авторизованих на даний момент часу користувачів прагне до нуля.

Таким чином, виникає поняття «сесії користувача» - при успішній авторизації на даного користувача відкривається так звана сесія, яка фактично є записом у вищезгаданому списку, користувачеві засобами мови програмування повідомляється тільки показчик на цей запис. При виконанні будь-яких операцій з даними користувач «повертає» цей показчик додатком, яке спочатку перевіряє, чи є у користувача за отриманим вказівником відкрита сесія і тільки потім виконує необхідні дії.

Кожному зареєстрованому користувачу виділяється свій особистий аккаунт, з якого він може додавати, видаляти і редагувати вже відправлені оголошення, а також підписуватися на поштову розсилку нових оголошень.

Підписка на поштову розсилку повинна проходити за загальноприйнятими правилами з підтвердженням активуючого коду, тобто повинна бути захищена від зловмисників.

Необхідно передбачити режим адміністрування з можливістю редагування основних параметрів системи працевлаштування а саме пошуку

оголошень, таких як назва, число оголошень, відображених на сторінку, включення або виключення поштової розсилки.

Адміністратор може створювати і видаляти розділи та підрозділи каталогу оголошень, переглядати список зареєстрованих користувачів, а так само видаляти користувачів.

2 АРХІТЕКТУРА СИСТЕМИ

2.1 Функціональна структура програми

Розглянемо основні завдання та вимоги на рівні організації WEB-інтерфейсу:

- інтерфейс відображення розділів каталогу та оголошень;
- інтерфейс аккаунта користувача;
- інтерфейс аккаунта адміністратора;
- інтерфейс підписки на поштову розсилку;
- інтерфейс авторизації і реєстрації користувачів.

На функціональному рівні:

- первісна інсталяція програми на сервері;
- з'єднання з БД MySQL;
- ініціалізація основних параметрів каталогу;
- перевірка на коректність значень змінних, прийнятих від користувача;
- висновок розділів і підрозділів каталогу, а також оголошень;
- реєстрація нових користувачів;
- авторизація користувачів;
- аутентифікація користувачів за допомогою механізму сесій і перевірка прав доступу;
- додавання, редагування та видалення оголошень;
- організація механізму поштової передплати із запитом підтвердження коду;
- підписка на підрозділи каталогу, активація, деактивація і видалення адреси з розсилки;
- розсилка оголошень на поштові адреси передплатників;

- редагування основних параметрів додатку;
- додавання і видалення користувачів;
- автоматичне видалення оголошень по закінченню терміну життя, видалення не продовжених сесій користувачів, видалення не активованих адрес поштової розсилки.

На рисунку 2.1 показана загальна схема програми та взаємодія між його основними частинами.

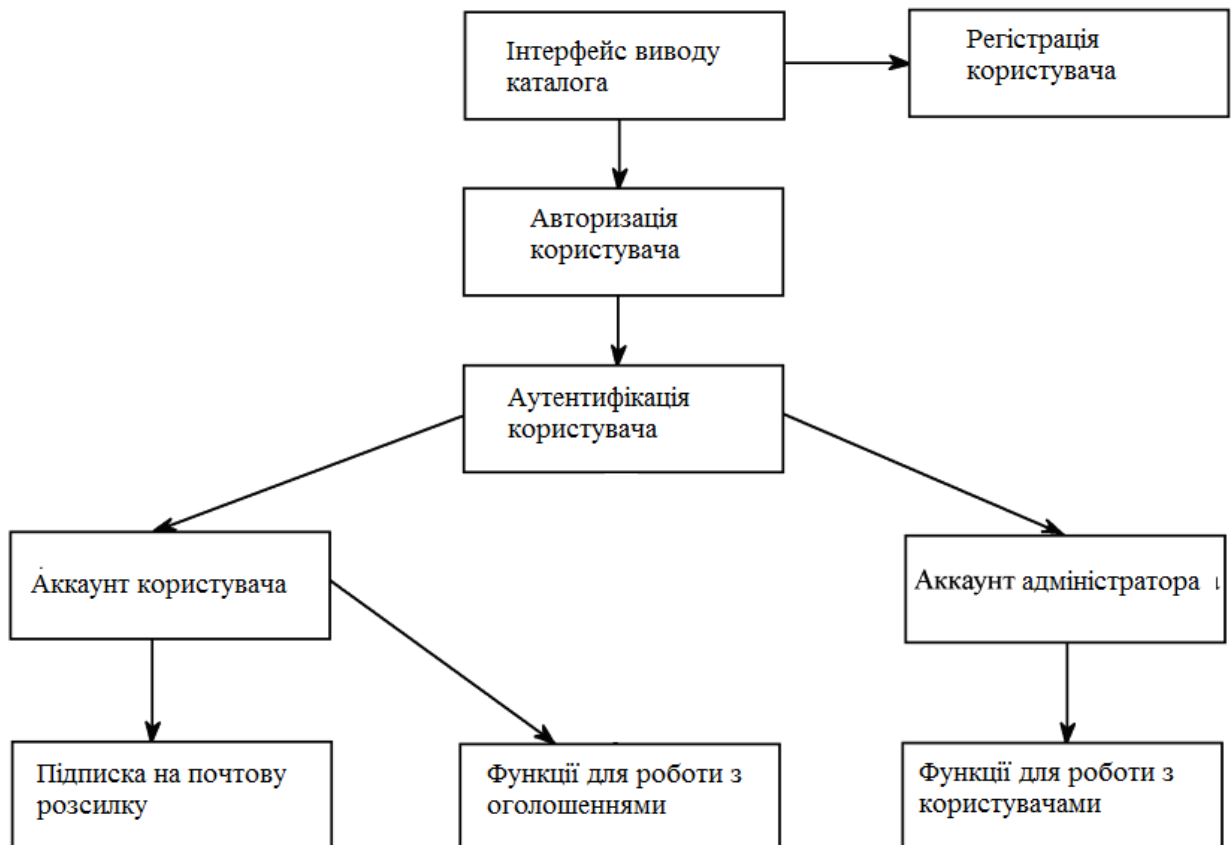


Рисунок 2.1 - Функціональна структура програми

Таким чином, проект доцільно реалізувати у вигляді декількох функціональних модулів, кожен з яких буде виконувати певну задачу:

- модуль інсталяції (з відображенням інтерфейсу);
- модуль з'єднання з БД MySQL;
- модуль відображення розділів каталогу та оголошень (з відображенням інтерфейсу);
- модуль реєстрації нових користувачів (з відображенням інтерфейсу);

- модуль авторизації користувачів (з відображенням інтерфейсу);
- модуль аутентифікації користувачів, заснованому на механізмі сесій;
- модуль реалізації аккаунта користувача (з відображенням інтерфейсу);
- модуль реалізації аккаунта адміністратора (з відображенням інтерфейсу);
- модуль реалізації аккаунта модератора (з відображенням інтерфейсу);
- модуль реалізації аккаунта підписки на поштову розсилку (з відображенням інтерфейсу).

2.2 Архітектура клієнт-сервер

Клієнт-серверна частина представляє собою обчислювальну або мережеву архітектуру, в якій завдання або мережева навантаження розподілені між постачальниками послуг, званими серверами, і замовниками послуг, званими клієнтами. Фізично клієнт і сервер - це програмне забезпечення. Зазвичай вони взаємодіють через комп'ютерну мережу за допомогою мережевих протоколів і знаходяться на різних обчислювальних машинах, але можуть виконуватися також і на одній машині. Програми - серверу, очікують від клієнтських програм запити і надають їм свої ресурси у вигляді даних (наприклад, завантаження файлів за допомогою HTTP, FTP (протокол передачі даних у мережі), BitTorrent, потокове мультимедіа або робота з базами даних) або сервісних функцій (наприклад, робота з електронною поштою, спілкування за допомогою систем миттєвого обміну повідомленнями, перегляд web-сторінок у всесвітній павутині).

Веб-застосунок отримує запит від клієнта і виконує обчислення, після цього формує веб-сторінку і відправляє її клієнтові мережею з використанням

протоколу HTTP. Саме веб-застосунок може бути клієнтом інших служб, наприклад, бази даних або стороннього веб-застосунку, розташованого на іншому сервері. Яскравим прикладом веб-застосунку є система управління вмістом статей Вікіпедії: безліч її учасників можуть брати участь у створенні мережевої енциклопедії, використовуючи для цього браузері своїх операційних систем (Microsoft Windows, GNU/Linux або будь-якої іншої операційної системи) без завантаження додаткових виконуваних модулів для роботи з базою даних статей.

Веб система з працевлаштування побудована на локальному сервері XAMPP Layered, та згодом впроваджена у глобальну мережу на сервері Apache, так як нашими основними критеріями є надійність, продуктивність та забезпечення безпеки, який містить у собі Apache сервер (рис. 2.1).

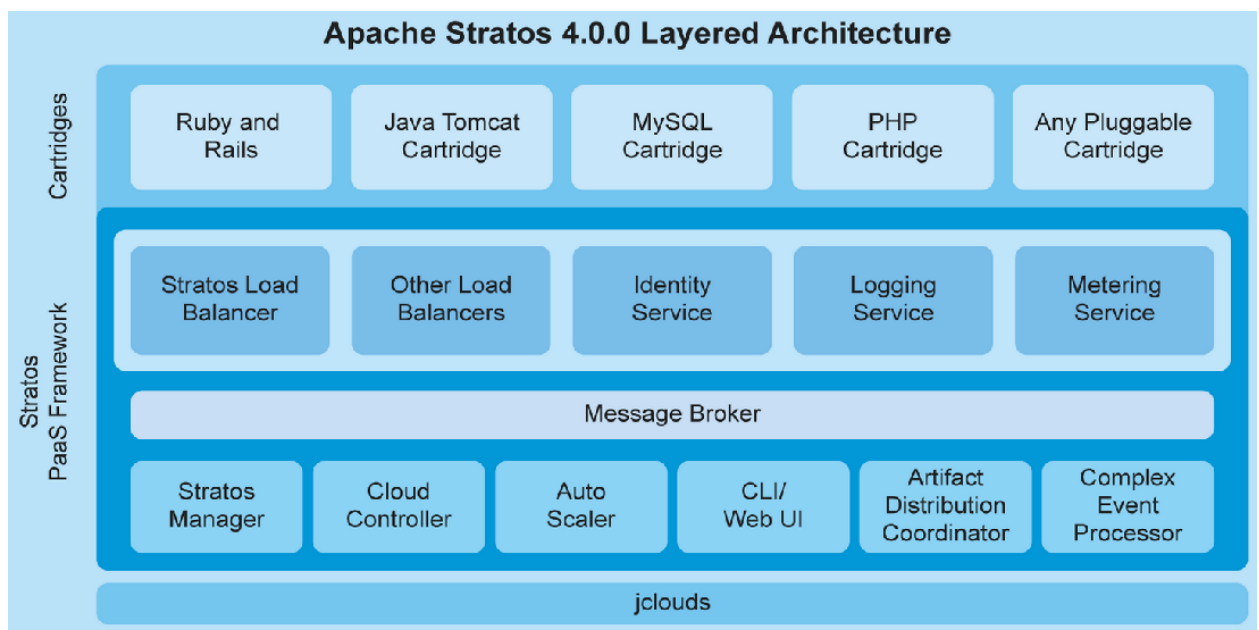


Рисунок 2.2 - Загальна архітектура Layered серверу

2.3 Функціональні модулі системи

Проект складається з 10 модулів, нижче буде описано кожен з них, а також функції, які в них використовуються.

1. Модуль `install.php`

Призначення: для початкового створення бази даних і таблиць, необхідних для роботи програми.

2. Модуль `tunes.php`

Призначення: містить константи зі значеннями логіна, пароля та імені бази даних, що використовуються для підключення до БД MySQL:

3. Модуль `index.php`

Призначення: відображення в гіпертекстовому форматі підрозділів і розділів каталогу та оголошень з поділом на сторінки.

4. Модуль `adduser.php`

Призначення: додавання нового облікового запису користувача.

5. Модуль `enter.php`

Призначення: авторизація користувача.

6. Модуль `sessions.php`

Призначення: аутентифікація користувача.

7. Модуль `users.php`

Призначення: відображення в гіпертекстовому форматі інтерфейсу аккаунта користувача і функціональний набір засобів для роботи з оголошеннями.

8. Модуль `mail.php`

Призначення: відображення в гіпертекстовому форматі інтерфейсу аккаунта підписки на поштову розсилку і необхідний функціональний набір засобів для роботи.

9. Модуль `moderator.php`

Призначення: відображення в гіпертекстовому форматі інтерфейсу аккаунта модератора і функціональний набір засобів для роботи з каталогом оголошень і видаленням повідомлень.

10. Модуль `admin.php`

Призначення: відображення в гіпертекстовому форматі інтерфейсу аккаунта адміністратора і функціональний набір засобів для роботи з користувачами, основними параметрами і розділами каталогу.

2.4 Основні функції системи

1. Алгоритм інсталяції.

Інсталяція – мається на увазі первісна установка програми на інтернет-сервер і створення аккаунта адміністратора. Після інсталяції адміністратор переходить в свій аккаунт і додає розділи та підрозділи в основний каталог.

Модуль-інсталятор виконує наступні дії:

- створює нову БД або видаляє всі таблиці в поточній БД, що використовуються додатком, якщо вони вже були створені;
- створює таблиці із зазначенням всіх необхідних полів і типів;
- додає записи зі значенням базових параметрів в статичні таблиці ACTION і OPTIONS;
- запитує логін і пароль адміністратора, додає запис в таблицю USERS і встановити права адміністратора.

2. Алгоритм відображення розділів і підрозділів.

Для відображення списку розділів необхідно надіслати запит на отримання всіх значень поля name з таблиці SUBJECT, де значення ідентифікатора розділу каталогу topic = 0, і вивести результат на екран.

Для відображення списку підрозділів використовується точно такий же метод, тільки значення ідентифікатора розділу каталогу topic порівнюється з поточним ідентифікатором розділу.

Для того щоб записи сортувалися в алфавітному порядку, всі запити доповнюються директивою «ORDER BY name ASC». Якщо в поточному підрозділі каталогу присутні оголошення декількох типів, виводиться

допоміжний список-фільтр з перерахуванням знайдених типів, тоді користувач може вибирати той чи інший тип оголошень, який буде подаватись на всіх розділах і підрозділах каталогу.

3. Алгоритм відображення оголошень із застосуванням фільтра на тип оголошень.

Для відображення оголошень беруться всі значення полів таблиці MESSAGES, в якій значення поля topic_id збігається зі значенням поточного ідентифікатора розділу каталогу. Оскільки при відображенні оголошень необхідно виводити ще й контактну інформацію про відправника, а також тип оголошення, в запит вводяться додаткові умови: значення поля user_id таблиці MESSAGES має збігатися зі значенням ідентифікатора користувача id таблиці USERS, а значення поля action_id - зі значенням поля id таблиці перерахування типів оголошень ACTION. Якщо користувач застосував фільтр на тип оголошень, запит доповнюється ще однією умовою: значення поля action_id повинно відповідати значенню фільтра. Сортування записів проводиться по полю id в порядку убутання, в результаті чого вгорі WEB-сторінки каталогу відображаються останні додані оголошення.

4. Алгоритм обмеження числа оголошень, виведених на одній WEB-сторінці.

Оскільки число оголошень, що задовольняють вищенаведені запитом може бути занадто велике, для того щоб не перевантажувати виводяться WEB-сторінки використовується алгоритм обмеження числа оголошень і формування так званої лінійки - посилань на сторінки, що містять результати запиту. Для цього за допомогою директиви COUNT попередньо підраховується кількість рядків у результуючому запиті, причому додаток запитує у БД саме число рядків в потрібному запиті, а не результат запиту, тим самим заощаджуючи час обміну даними і трафік (у разі якщо БД знаходиться на іншому віддаленому сервері) . За отриманого значення визначається, чи потрібно формувати лінійку чи ні. Якщо потрібно, запит доповнюється директивою LIMIT з перерахуванням діапазону виведених значень в

результуючому запиті. Діапазон являє собою деяке початкове значення покажчика на «вікно» і максимальний розмір цього «вікна». Таким чином, результат запити ділиться на «вікна» і не перевантажує виводяться сторінки. Користувач може вибирати для перегляду те чи інше «вікно», при цьому сам зміст запити не змінюється, змінюються лише межі діапазону.

5. Алгоритм передачі значення фільтра при переходах по посиланнях каталогу.

Включивши фільтр на тип оголошень, користувач може перейти з поточної сторінки каталогу на яку-небудь іншу, причому під час таких переходів додаток повинен «знати», що фільтр був включений. Найоптимальніший спосіб для збереження значення фільтра в разі програмування мовою PHP - це використання механізму сесій.

6. Алгоритм реєстрації нових користувачів.

Реєстрація користувача відбувається в наступній послідовності: на запит програми користувач вводить логін, пароль та контактну інформацію про себе або про організацію, яку він представляє на дошці оголошень. Для зручності роботи адміністратора зі списком логінів користувачів проводиться перевірка на припустимий набір символів, що складається тільки з англійських букв і цифр. Для того щоб виключити реєстрацію користувачів з однаковими логінами, необхідно зробити перевірку по полю login таблиці USERS. Після таких перевірок обчислюється хеш-код пароля користувача. Хеш-код являє собою 32-х символну послідовність, обчислену за алгоритмом «MD5 Message-Digest Algorithm» (розробка корпорації RSA Data Security, [3]). Отримана послідовність є унікальною для символного рядка пароля користувача, тобто ймовірність того, що два однакових пароля дадуть однаковий хеш-код, прагне до нуля. Основна перевага кодування за алгоритмом MD5 - це неможливість зворотного відновлення первісної символного рядка (пароля) за отриманим хеш-коду. Вся отримана інформація додається в поля login і password таблиці USERS, значення поля access встановлюється рівним 1, що означає «права доступу на рівні користувача».

7. Алгоритм авторизації користувачів.

Користувачеві виводиться форма для запиту логіна і пароля, після введення значень обчислюється хеш-код пароля і надсилається запит до таблиці USERS. Якщо у користувача є обліковий запис, відповідна цим значенням, то він вважається авторизованим. Прочитавши значення поля id, ми отримаємо ідентифікатор користувача. Потім необхідно подбати про інформацію для модуля аутентифікації, для цього використовується таблиця сесій SESSION. Поле user_id цієї таблиці містить значення ідентифікаторів користувачів з відкритими сесіями. Оскільки на один і той же логін може бути відкрита тільки одна сесія, спочатку з таблиці видаляється запис з отриманим раніше значенням ідентифікатором авторизованого користувача. Далі випадковим чином генерується 20-символьний рядок, яку позначимо як «код сесії авторизованого користувача». Саме цей рядок, а не хеш-код пароля користувача (або тим більше його справжній пароль) згодом буде передаватися між додатком на сервері і браузером клієнта. Код сесії заноситься в поле sid. В цілях безпеки, для того щоб відкриті сесії мали обмежений в часі термін дії, кожній сесії призначається час життя time, рівне поточного значення часу на сервері плюс 60 хвилин. У випадку, якщо користувач ввів помилкові дані і його обліковий запис була виявлено, йому пропонується або заново пройти процедуру авторизації, або зареєструвати новий обліковий запис.

8. Алгоритм аутентифікації користувачів.

Звернення до даних йде через запити до таблиці сесій SESSIONS і таблиці користувачів USERS. Спочатку йде спроба вважати 20-х символьний хеш-код сесії авторизованого користувача, що передається між додатком і авторизованим клієнтом. Потім з таблиці SESSIONS видаляються всі сесії з простроченим часом життя, якщо такі є. Для цього поточне значення часу на сервері порівнюється зі значенням поля time.

З таблиці SESSIONS отримуємо значення id того запису, у якій значення sid збігається з хеш-кодом сесії авторизованого користувача, після чого з

таблиці USERS вибирається обліковий запис з отриманим ідентифікатором користувача id. Цей обліковий запис містить ідентифікатор користувача, логін, контактну інформацію та права доступу, яка в подальшому використовується різними модулями, які вимагають аутентифікацію.

Далі продовжується термін життя сесії: оновлюється значення поля time.

У тому випадку, якщо 20-х символний хеш-код сесії авторизованого користувача вважати не вдалося, або така сесія була знайдена в таблиці SESSIONS, то аутентифікація не можлива і користувач вважається неавторизований.

Якщо користувач закриває сесію самостійно, відбувається негайне видалення запису з таблиці SESSIONS.

9. Алгоритм додавання нових оголошень.

Спочатку йде перевірка максимально допустимої довжини оголошення, значення якої зберігається в одній із запису таблиці OPTIONS. Оголошення коротше п'яти символів також вважаються неприпустимими. Оскільки повідомлення може містити неприпустимі HTML-теги, відбувається заміна таких тегів на їх «безпечні» еквіваленти.

З таблиці MESSAGES видаляються повідомлення, у яких закінчився термін життя, потім додається новий запис зі значенням полів topic_id, рівному ідентифікатору поточного підрозділу каталогу оголошень в таблиці SUBJECT; user_id, що вказує на ідентифікатор користувача, що відправив оголошення в таблиці USERS; action_id, що вказує на ідентифікатор типу повідомлення в таблиці ACTION; time, рівний значенню поточної дати; time_live, що позначає термін життя оголошення, і text, що містить текст самого оголошення.

У разі невиконання умов перевірок оголошення в БД не заноситься.

10. Алгоритм управління підпискою на підрозділи каталогу оголошень.

В поля mail_id і topic_id таблиці MAILSUB нового запису заносяться відповідні значення ідентифікатора запису передплатника в таблиці MAIL, а також підрозділу каталогу, на який оформляється підписка. У разі якщо

підписка вже оформлена, нова запису не створюється. Якщо користувач хоче відписатися від якого-небудь підрозділу, запис видаляється.

11. Алгоритм поштової розсилки оголошень.

Для того щоб отримати список поштових адрес, на які оформлена поштова розсилка для поточного підрозділу каталогу оголошень, потрібно з таблиці MAIL вибрати значення поля email тих записів, в яких значення поля id збігається зі значенням поля mail_id таблиці MAILSUB і значення поля topic_id одно ідентифікатору поточного підрозділу каталогу. При цьому значення поля activation таблиці MAIL має дорівнювати «1».

За отриманого списку організується цикл, в якому на кожен адресу відправляється текст оголошення.

Блок-схеми основних алгоритмів наведені на рисунку 2.2.

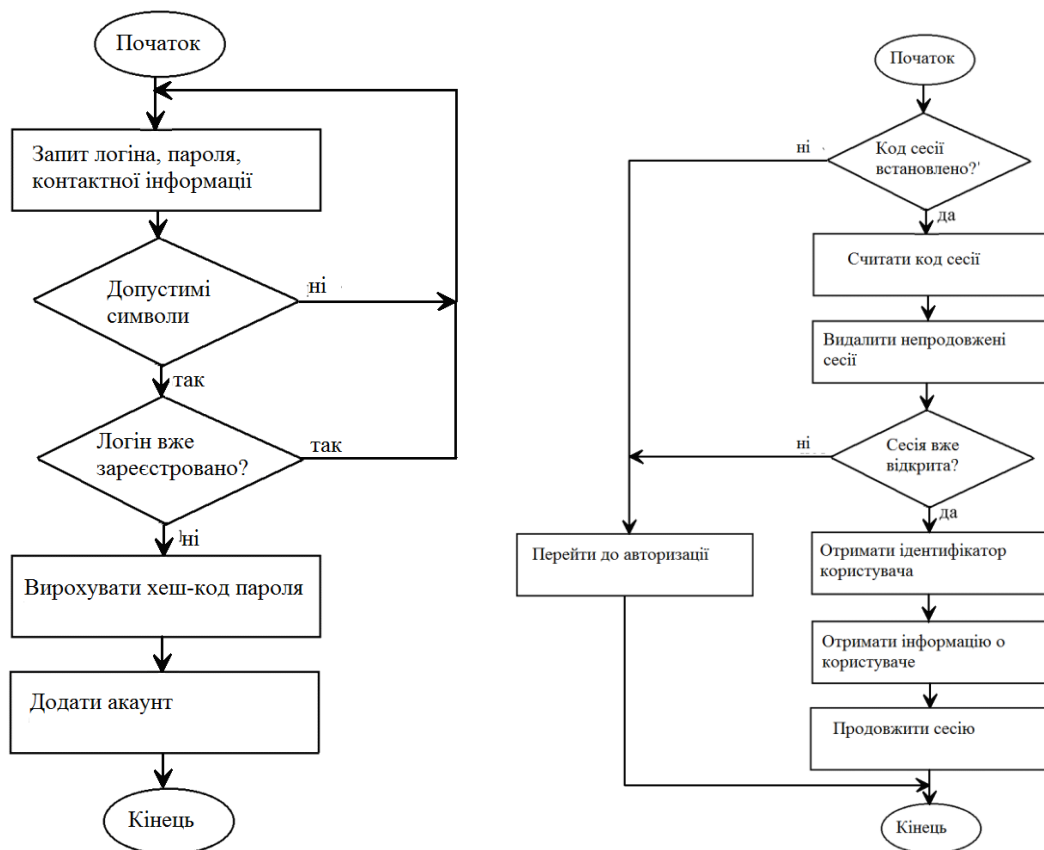


Рисунок 2.3 - Блок-схеми алгоритмів реєстрації та авторизації

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЇЇ АНАЛІЗ

3.1 Застосовані мови програмування

PHP (PHP гіпертекстовий препроцесор) - скриптова мова програмування, була створена для генерації HTML-сторінок (стандартна мова розмітки веб-сторінок в Інтернеті) на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP - проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта.

PHP - мова, яка може бути вбудована безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP -інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустрине парну екрануючу послідовність (?>).

HTML5 - мова для структурування та подання вмісту всесвітньої павутини. Це п'ята версія HTML. Хоча стандарт був завершений (рекомендована версія до використання) тільки в 2014 році (попередня, четверта, версія опублікована в 1997 році [8]), ще з 2013 року [9] браузерами оперативно здійснювалася підтримка, а розробниками - використання робочого стандарту (англ. HTML Living Standard). Мета розробки HTML5 - поліпшення рівня підтримки мультимедіа-технологій з одночасним збереженням удобочитаємості коду для людини і простоти аналізу для парсерів.

CSS3 (англ. Cascading Style Sheets 3 [6] - каскадні таблиці стилів третього покоління) - активно розробляється специфікація CSS [7]. Являє собою формальну мову, реалізований за допомогою мови розмітки. Наймасштабніша

редакція порівняно з CSS1, CSS2 і CSS2.1. Головною особливістю CSS3 є можливість створювати анімовані елементи без використання JS, підтримка лінійних і радіальних градієнтів, тіней, згладжування і багато іншого.

Переважно використовується як засіб опису та оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватися до будь XML-документів, наприклад, до SVG або XUL.

3.2 Проектна База даних

Задля забезпечення максимальної швидкості обміну даними між серверною частиною та клієнтом у якості бази даних було обрано реляційну СУБД PhpMyAdmin з базою даних MySQL.

MySQL - вільна реляційна система управління базами даних. Розробку і підтримку MySQL здійснює корпорація Oracle, що отримала права на торговельну марку разом з поглиненої Sun Microsystems, яка раніше придбала шведську компанію MySQL AB [4]. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів. Саме завдяки такому замовленню майже в найраніших версіях з'явився механізм реплікації.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, AppServ, LAMP і в портативні збірки серверів Денвер, ХАМРР, VertrigoServ. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або видалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми [5].

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць.

Розглянемо призначення і структуру таблиць, використовуваних у проєкті:

1. Таблиця OPTIONS.

Статична таблиця, призначена для зберігання основних параметрів оголошень, складається з трьох полів:

id	name	value
----	------	-------

id - порядковим номер запису, тип поля smallint (допустиме значення до 32767), ключове.

name - назва параметра, тип поля text (до 65535 символів)

value - значення параметра, тип text.

У даній версії проєкту в таблиці міститься шість записів, які заносяться при інсталяції. Зміст записів поля name: назва BBS, число відображених на сторінку оголошень, максимальний час життя оголошення, розсилка оголошень поштою, видалення оголошень по закінченню часу життя, максимальний розмір оголошення.

При необхідності адміністратор може змінити будь-яке з початкових значень поля value.

2. Таблиця ACTION.

Статична таблиця, містить тип оголошень, складається всього з двох полів:

id	action
----	--------

id - порядковим номер запису, тип поля smallint, ключове.

action - назва типу оголошення, тип поля text.

Зміст записів поля action: пропозиція, попит, обмін, оренда, інше. Значення полів задаються автоматично в процесі інсталяції і в наступному часу не змінюються.

3. Таблиця SUBJECT.

Призначена для зберігання індексу основного каталогу (містить інформацію про структуру розділів і підрозділів, що утворюють каталог).

id	topic	name
----	-------	------

id - ідентифікатор розділу або підрозділу, тип поля int (значення до 2147483647), ключове. Для того щоб даний ідентифікатор був унікальним (з неповторяючимися значеннями), полю призначений додатковий тип auto_increment. Під час першого створення таблиці значення цього поля дорівнює одиниці, при додаванні нового запису його значення автоматично інкрементується. Оскільки видалення записів з таблиці не впливає на значення цього лічильника, ми отримуємо унікальність ідентифікатора запису. Для оптимізації пошуку по таблиці встановлюємо тип поля index.

topic - значення ідентифікатора розділу каталогу, тип поля int, ключове. Якщо даний запис описує не підрозділ, а кореневий розділ каталогу, то значення поля дорівнює 0.

name - назва розділу або підрозділу каталогу, тип поля text.

Оскільки адміністратор каталогу може додавати і видаляти розділи та підрозділи, число записів у цій таблиці не постійно.

4. Таблиця USERS.

Містить інформацію про зареєстрованих користувачів.

id	login	password	contact	access
----	-------	----------	---------	--------

id - ідентифікатор користувача, тип поля int, ключове, auto_increment, index.

login - логін користувача, тип поля text.

password - 32-х символний хеш-код пароля користувача, тип поля text.

contact - контактна інформація користувача, тип поля text.

access - права доступу, тип поля smallint. 0 - адміністратор, 1 - користувач, 2 - модератор.

5. Таблиця MESSAGES.

Призначена для зберігання тексту оголошень і їх параметрів, заданих відправником.

id	topic_id	user_id	action_id	time	time_live	text
----	----------	---------	-----------	------	-----------	------

id - ідентифікатор оголошення, тип поля int, ключове, auto_increment, index.

topic_id - значення ідентифікатора підрозділу каталогу, тип поля int, index.

user_id - значення ідентифікатора відправника (користувача), тип поля int.

action_id - значення ідентифікатора типу оголошення, тип поля smallint.

time - дата написання оголошення, тип поля text.

time_live - час видалення оголошення в UNIX-форматі, тип поля bigint.

text - текст оголошення, тип поля text.

6. Таблиця MAIL.

Містить адреси та параметри, що використовуються для поштової розсилки оголошень.

id	user_id	email	activation	time
-----------	----------------	--------------	-------------------	-------------

id - ідентифікатор запису, тип поля `int`, ключове, `auto_increment`, `index`.

user_id - значення ідентифікатора відправника (користувача), тип поля `int`.

email - поштову адресу передплатника на розсилку, тип поля `text`.

activation - код для активації, тип поля `text`. Після підтвердження активації значення поля дорівнює 1. При тимчасовому виключенні розсилки значення поля 0.

time - містить значення часу в UNIX-форматі, по закінченню якого не активована запис буде автоматично видалена з таблиці, тип поля `bigint`.

7. Таблиця MAILESUB.

Описує зв'язок між підрозділами та адресами передплатників, яка використовується для поштової розсилки оголошень.

mail_id	topic_id
----------------	-----------------

mail_id - значення ідентифікатора запису передплатника в таблиці MAIL, тип поля `int`, `index`.

topic_id - значення ідентифікатора підрозділу, на який оформлена передплата, тип поля `int`, `index`.

8. Таблиця SESSIONS.

Містить значення ідентифікаторів сесій авторизованих користувачів, використовується для реалізації механізму безпечної аутентифікації.

user_id	sid	time
----------------	------------	-------------

user_id - значення ідентифікатора користувача в таблиці USERS, тип поля `int`.

sid - ідентифікатор сесії авторизованого користувача, тип поля `text`.

time - значення часу в UNIX-форматі, по закінченню якого не продовжена сесія буде автоматично видалена з таблиці, тип поля `bigint`.

Зв'язки між полями таблиць наведені на рисунку 3.1 структурної схеми даних. Ключові поля-ідентифікатори позначені знаком "*", типи використуваних зв'язків: "один до багатьох" і "один до одного".

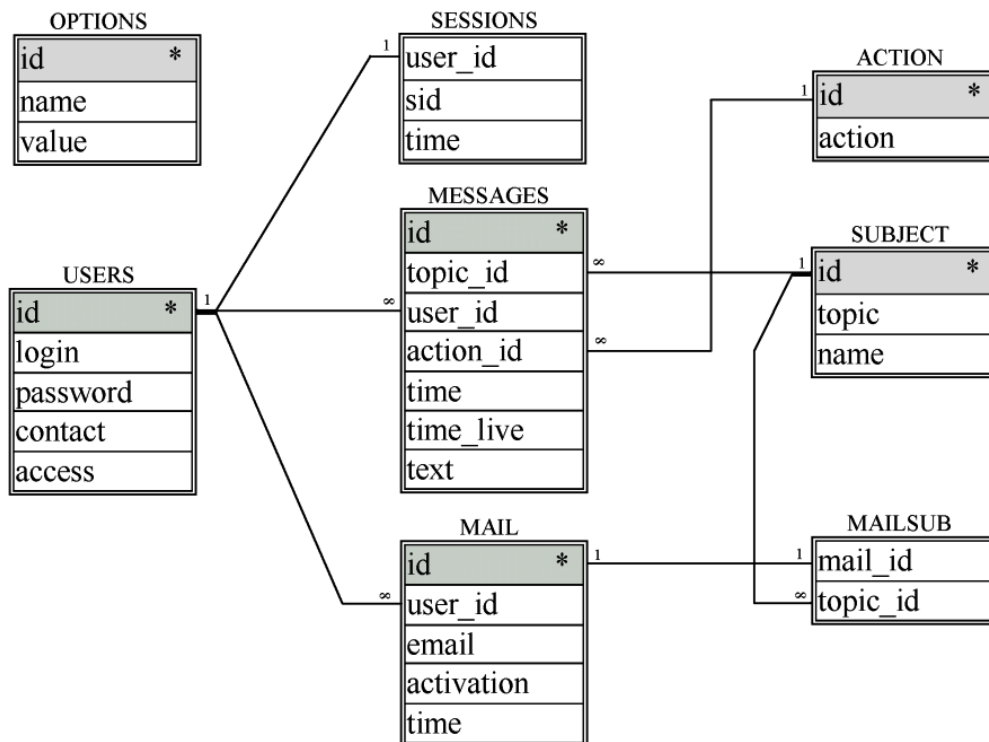


Рисунок 3.1 - Схема даних

3.3 Допоміжні технології Laravel

Laravel - безкоштовний веб - фреймворк з відкритим кодом , призначений для розробки з використанням архітектурної моделі MVC . Laravel випущений під ліцензією MIT . В результаті опитування sitepoint.com в грудні 2013 про найпопулярніші PHP - фреймворк Laravel зайняв місце самого багатообіцяючого проекту на 2014 рік [11].

У 2015 році в результаті опитування з використання PHP - фреймворк серед програмістів посів перше місце в номінаціях:

- фреймворк корпоративного рівня;

- фреймворк для особистих проєктів.

3.4 Допоміжні технології - JQuery

jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елемента DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з AJAX. Зараз розробка jQuery ведеться командою jQuery на чолі з Джоном Резігом.

Можливості:

- движок кросбраузерності CSS-селекторів Sizzle [8], що виділився в окремий проєкт;
- перехід по дереву DOM, включаючи підтримку XPath як плагіна;
- події;
- візуальні ефекти;
- AJAX-доповнення;
- JavaScript - плагіни.

3.5 Опис програмної реалізації

Додаток встановлюється на сервері, що підтримує виконання PHP-скриптів не нижче третьої версії. Для роботи програми з даними необхідна СУБД MySQL версії 3.23 або вище, яка з метою максимальної продуктивності

повинна бути встановлена на тому ж сервері, що і інтерпретатор мови PHP, однак ця умова не є обов'язковим.

Для установки програми необхідно по FTP-протоколу або будь-яким іншим чином скопіювати файли з поставляється дистрибутива в каталог виділеного сервера, прописати назву БД, логін і пароль для доступу (видаються адміністратором інтернет-сервера) в модулі `tunes.php` і запустити інсталятор `install.exe`. Після введення реєстраційної інформації адміністратора система з працевлаштування та ведення бізнесу відразу ж готова до роботи.

Перегляд оголошень здійснюється через інтуїтивно-зрозумілий WEB-інтерфейс.

Для зручності користувача, всі оголошення розділені на тематики, і містяться в упорядкованому порядку в розділах каталогу. Кожному оголошенню відповідає тип - «попит», «пропозиція», «інше». Користувач може включити фільтр на відображення оголошень тільки потрібного йому типу.

Оголошення можуть лише зареєстровані користувачі. При реєстрації у користувача запитується логін, пароль і контактна інформація. Після реєстрації користувач може працювати у своєму акаунті, додавати, видаляти і редагувати вже відправлені оголошення, а також підписуватися на поштову розсилку нових оголошень.

Механізм підписки на поштову розсилку має простий і зручний інтерфейс, захищений від зловмисників, що намагаються підписати чийсь чужий поштову скриньку. Кожне доданий оголошення розсилається на поштові адреси передплатників без будь-яких штучних обмежень і тимчасових затримок, тим самим забезпечуючи ефективність інформування.

Передбачений режим адміністрування з можливістю редагування основних параметрів оголошень, таких як назва, число оголошень, відображених на сторінку, включення або виключення поштової розсилки.

Адміністратор може створювати і видаляти розділи та підрозділи каталогу оголошень, переглядати список зареєстрованих користувачів, а так само видаляти користувачів.

Авторизація користувачів має простий інтерфейс і заснована на безпечному алгоритмі аутентифікації, який забезпечує надійний захист від злому паролів користувачів для додатків подібного рівня.

3.6 Ролі та права користувачів

У даній веб системі передбачено розширення користувачів на певні ролі у системі. Таким чином, ми домоглися чіткого розмежування прав користувачів у нашій системі, завдяки чому збільшилася секьюрність системи. У кожній ролі в системі є свої особисті привілеї. Нижче приведені деякі з них:

1. Гість (користувач, який ще не зареєструвався у системі):

- можливість зареєструватися як здобувач або працевлаштувач через форму;
- може переглянути вакансії та резюме.

2. Здобувач (зареєстрований користувач з правами розміщення та подачі резюме на вакансію):

- можливість редагування реєстраційної інформації;
- можливість зареєстрування резюме;
- перегляд особистого резюме з правами його редагування;
- e-mail підписка на вакансії та подальший перегляд;
- перегляд вакансій та інших резюме.

3. Роботодавець (зареєстрований користувач з правами розміщення вакансій):

- можливість редагування реєстраційної інформації;

- можливість зареєстрування компанії та вакансій;
- можливість редагування компанії та розміщених вакансій;
- e-mail підписка на резюме та подальший перегляд;
- перегляд інших вакансій та інших резюме;
- пошук співробітника за критеріями;
- перегляд архіву резюме.

4. Адміністратор:

- додавання/перегляд/редагування/видалення резюме;
- додавання/перегляд/редагування/видалення вакансій;
- додавання/перегляд/редагування/видалення здобувачів;
- додавання/перегляд/редагування/видалення роботодавців;
- додавання/перегляд/редагування/видалення категорій;
- додавання/перегляд/редагування/видалення регіону;
- видалення дублікатів e-mail повідомлень з БД;
- пошук по заданим критеріям для здобувачів, роботодавців, вакансій, резюме, категорій, регіонам тощо.

3.7 Зовнішній вигляд веб системи

Інтерфейс програми виглядає наступним чином:

Вход Регистрация

Получать вакансии Разместить резюме

Найти работу + Разместить резюме Работодателю

Поиск вакансий

Например: курьер Харьков Город (до трех) Все категории Категории (до трех) весь период Актуальность **НАЙТИ**

Рисунок 3.2 – Головна сторінка сайту

Задля поліпшення пошуку вакансії та резюме було сформовано певний фільтр за категоріями та регіонами (рис. 3.3, 3.4).

Работа в Харькове

Вакансии в Харькове по категориям

HR-специалисты/бизнес-тренеры	IT-компьютеры/Интернет
Банки/Финансы	Бухгалтерия/Аудит
Гостинично-ресторанное дело	Домашний персонал
Жилищное-коммунальное хозяйство	Инженеры/Технологи/Проектировщики
Красота/Спорт/Туризм	Логистика/ВЭД
Маркетологи/Специалисты по PR	Медицина/Фармацевтика
Недвижимость	Образование/Наука
Охрана/Безопасность	Переводчики
Полиграфия	Производство
Работа для пенсионеров / людей с ограниченными возможностями	Работа для студентов
Работа на дому	Работа за рубежом
Рабочие/Сборщики	Работа на лето
Реклама	Разное
Сельское хозяйство	Секретари/Офис-менеджеры
СМИ	Сетевой маркетинг/Работа в Интернет
Строительство	Страхование
Телекоммуникации/Связь	Сфера обслуживания
Торговля/Продажи	ТОП-менеджмент
Шоу-бизнес	Транспорт/Автосервис
	Юриспруденция

Рисунок 3.3 - Виведення всіх видів вакансій

Поиск работы по регионам		
Работа в Александрии	Работа в Керчи	Работа в Павлограде
Работа в Алчевске	Работа в Киеве	Работа в Полтаве
Работа в Белой Церкви	Работа в Кировограде	Работа в Ровно
Работа в Бердянске	Работа в Краматорске	Работа в Севастополе
Работа в Виннице	Работа в Кременчуге	Работа в Северодонецке
Работа в Горловке	Работа в Кривом Роге	Работа в Симферополе
Работа в Днепродзержинске	Работа в Крыму	Работа в Сумах
Работа в Днепропетровске	Работа в Луганске	Работа в Тернополе
Работа в Донецке	Работа в Луцке	Работа в Ужгороде
Работа в Дрогобыче	Работа в Львове	Работа в Феодосии
Работа в Евпатории	Работа в Мариуполе	Работа в Харькове
Работа в Житомире	Работа в Мелитополе	Работа в Херсоне
Работа в Закарпатье	Работа в Мукачево	Работа в Хмельницком
Работа в Запорожье	Работа в Николаеве	Работа в Черкассах
Работа в Ивано-Франковске	Работа в Никополе	Работа в Чернигове
Работа в Измаиле	Работа в Новомосковске	Работа в Черновцах
Работа в Каменце-Подольском	Работа в Одессе	Работа в Ялте

Рисунок 3.4 – Виведення всіх назв регіону

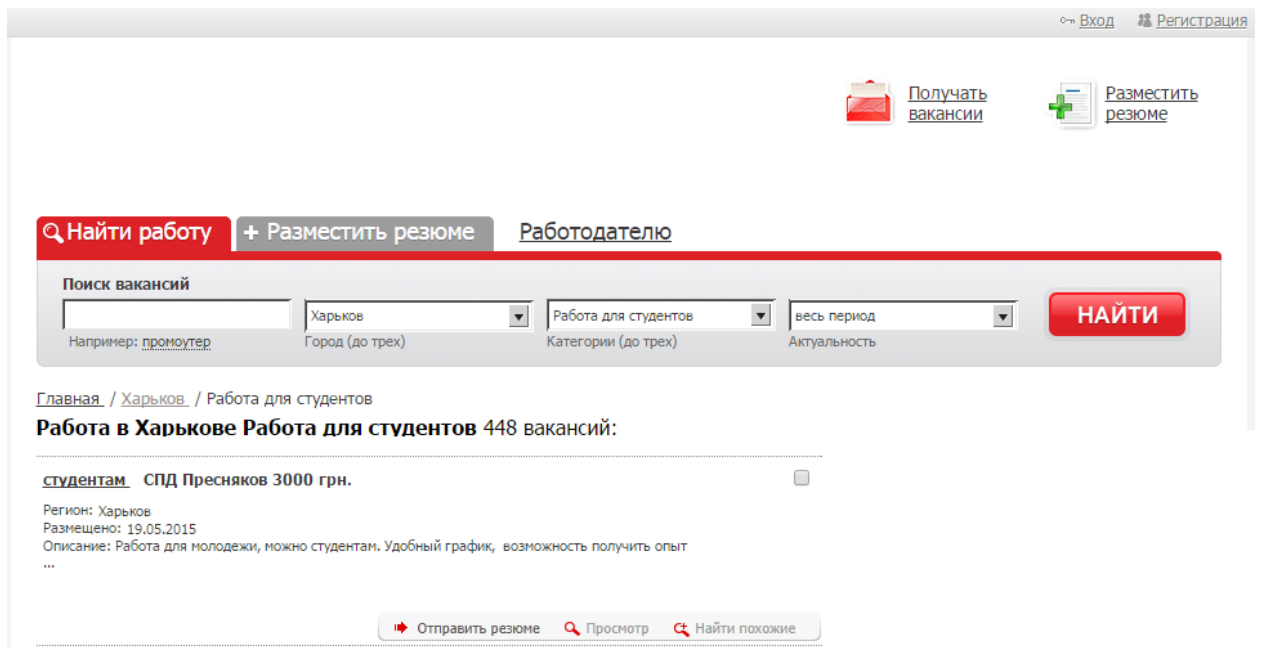


Рисунок 3.5 - Висновок вакансій із застосуванням фільтра на категорію «Харків», «Робота для студентів»

У даній веб системі передбачено розшарування користувачів на певні ролі у системі. Таким чином, ми домоглися чіткого розмежування прав користувачів у нашій системі (рис. 3.6, 3.7).

Регистрация

Соискатель Работодатель

* Обязательные поля

E-mail *

Пароль *

Показать пароль

Ф.И.О. *

Телефон

Получать новости на мой e-mail


Регистрация 

Рисунок 3.6 - Реестрация нового користувача у ролі здобувача

Регистрация

Соискатель
 Работодатель

* Обязательные поля

E-mail *
 Пароль *
 Показать пароль

Название компании *
 Прямой работодатель
 Кадровое агентство

Отрасль *
 Город *
 Сайт

Контактная информация для менеджеров

не отображать на сайте

Ф.И.О. *
 Телефон *

Получать новости на мой e-mail

Регистрация >

Рисунок 3.7 – Реестрация нового користувача у ролі роботодавця

Для процесу авторизації зареєстрованого користувача була реалізована форма (рис. 3.8).

[Вход](#)
[Регистрация](#)

[Забыли пароль?](#)

Рисунок 3.8 - Авторизація користувача

Задля оформлення підписки на e-mail потрібно вибрати категорії та певний регіон (рис. 3.9) за допомогою якої можна знайти існуючу вакансію чи резюме.

- 1** Выберите интересующие вас категории и города, вакансии из которых вы хотите получать на почту. Вы также можете указать ключевое слово (название должности). Количество выбранных категорий и городов не ограничено.

Категория *

<input type="checkbox"/> HR-специалисты/бизнес-тренеры	<input type="checkbox"/> IT-компьютеры/Интернет	<input type="checkbox"/> Банки/Финансы
<input type="checkbox"/> Бухгалтерия/Аудит	<input type="checkbox"/> Гостинично-ресторанное дело	<input type="checkbox"/> Домашний персонал
<input type="checkbox"/> Жилищное-коммунальное хозяйство	<input type="checkbox"/> Инженеры/Технологи/Проектировщики	<input type="checkbox"/> Красота/Спорт/Туризм
<input type="checkbox"/> Логистика/ВЭД	<input type="checkbox"/> Маркетологи/Специалисты по PR	<input type="checkbox"/> Медицина/Фармацевтика
<input type="checkbox"/> Недвижимость	<input type="checkbox"/> Образование/Наука	<input type="checkbox"/> Охрана/Безопасность
<input type="checkbox"/> Переводчики	<input type="checkbox"/> Полиграфия	<input type="checkbox"/> Производство
<input type="checkbox"/> Работа для пенсионеров / людей с ограниченными возможностями	<input type="checkbox"/> Работа для студентов	<input type="checkbox"/> Работа за рубежом
<input type="checkbox"/> Рабочие/Сборщики	<input type="checkbox"/> Работа на дому	<input type="checkbox"/> Работа на лето
<input type="checkbox"/> Секретари/Офис-менеджеры	<input type="checkbox"/> Разное	<input type="checkbox"/> Реклама
<input type="checkbox"/> СМИ	<input type="checkbox"/> Сельское хозяйство	<input type="checkbox"/> Сетевой маркетинг/Работа в Интернет
<input type="checkbox"/> Сфера обслуживания	<input type="checkbox"/> Страхование	<input type="checkbox"/> Строительство
<input type="checkbox"/> Торговля/Продажи	<input type="checkbox"/> Телекоммуникации/Связь	<input type="checkbox"/> ТОП-менеджмент
<input type="checkbox"/> Юриспруденция	<input type="checkbox"/> Транспорт/Автосервис	<input type="checkbox"/> Шоу-бизнес

Город *

<input type="checkbox"/> Вся Украина	<input type="checkbox"/> Александрия	<input type="checkbox"/> Алчевск
<input type="checkbox"/> Белая Церковь	<input type="checkbox"/> Бердянск	<input type="checkbox"/> Борисполь
<input type="checkbox"/> Винница	<input type="checkbox"/> Горловка	<input type="checkbox"/> Днепродзержинск
<input type="checkbox"/> Днепропетровск	<input type="checkbox"/> Донецк	<input type="checkbox"/> Дрогобыч
<input type="checkbox"/> Евпатория	<input type="checkbox"/> Житомир	<input type="checkbox"/> Закарпатье
<input type="checkbox"/> Запорожье	<input type="checkbox"/> Ивано-Франковск	<input type="checkbox"/> Измаил
<input type="checkbox"/> Каменец-Подольский	<input type="checkbox"/> Керчь	<input type="checkbox"/> Киев
<input type="checkbox"/> Кировоград	<input type="checkbox"/> Краматорск	<input type="checkbox"/> Кременчуг
<input type="checkbox"/> Кривой Рог	<input type="checkbox"/> Крым	<input type="checkbox"/> Луганск
<input type="checkbox"/> Луцк	<input type="checkbox"/> Львов	<input type="checkbox"/> Мариуполь
<input type="checkbox"/> Мелитополь	<input type="checkbox"/> Мукачево	<input type="checkbox"/> Николаев
<input type="checkbox"/> Никополь	<input type="checkbox"/> Новомосковск	<input type="checkbox"/> Одесса
<input type="checkbox"/> Павлоград	<input type="checkbox"/> Полтава	<input type="checkbox"/> Ровно
<input type="checkbox"/> Севастополь	<input type="checkbox"/> Северодонецк	<input type="checkbox"/> Симферополь
<input type="checkbox"/> Сумы	<input type="checkbox"/> Тернополь	<input type="checkbox"/> Ужгород
<input type="checkbox"/> Феодосия	<input type="checkbox"/> Харьков	<input type="checkbox"/> Херсон
<input type="checkbox"/> Хмельницкий	<input type="checkbox"/> Черкассы	<input type="checkbox"/> Чернигов
<input type="checkbox"/> Черновцы	<input type="checkbox"/> Ялта	

* Поля обязательные для заполнения

- 2** Нажмите кнопку "Подписаться"



Рисунок 3.9 - Аккаунт користувача

Для створення нового оголошення була реалізована форма (рис. 3.10, 3.11, 3.12)

Информация о вакансии
* Обязательные поля

Название вакансии *

Например: Программист 1С, Менеджер по продажам, Главный бухгалтер

Категория размещения вакансии *

- HR-специалисты/Бизнес-тренеры
- IT-компьютеры/Интернет
- Банки/Финансы
- Бухгалтерия/Аудит
- Гостинично-ресторанное дело
- Домашний персонал

Город *

- Александрия
- Ачехос
- Белая Церковь
- Бердянск
- Борисполь
- Вишниця

Занятость *

Зароботна плата

от до Валюта грн.

комментарий к зарплате

Опыт работы *

Уровень образования *

Возраст

от до

Описание вакансии *

Требования:
Обязанности:
Условия труда:

B *I* U **ABC**

Максимум 1000 символов (осталось 3964)

Контактная информация

не отображать на сайте

Ф.И.О.

Телефон

E-mail *

Получать резюме Вашего направления на nextdemx1@gmail.com

Сохранить или [Отменить](#)

Рисунок 3.10 - Додавання нової вакансії від роботодавця

Создать резюме ➔ Шаг 1 ➔ Шаг 2

Персональные данные

Ф. И. О.* конфиденциально

Год рождения*

Город* Вся Украина
 Александрия
 Алчевск
 Белая Церковь
 Бердянск
 Борислав

Название должности, на которую претендуете*

Категория* HR-специалисты/бизнес-тренеры
 IT-компьютеры/Интернет
 Банки/Финансы
 Бухгалтерия/Аудит
 Гостинично-ресторанное дело
 Домашний персонал

E-mail*

Телефон

Ваше фото: Файл не выбран

* Поля обязательные для заполнения



Рисунок 3.11 – Створення резюме (етап 1)

Создать резюме → Шаг 1 → **Шаг 2**

Образование

Уровень образования* высшее ▼

Добавление места учебы

Период обучения* с ▼ по ▼

Учебное заведение*

Специальность*

Опыт работы

Опыта работы не имею

Добавление места работы

Период работы* с ▼ ▼ по настоящее вр ▼ ▼

Название компании*

Сфера деятельности* ▼

Должность*

Обязанности, достижения

Добавить ещё

Дополнительная информация

B
I
U
ABC
↺
↻
✎
☰
☰

Не показывать это резюме в результатах поиска

Создать подписку на получение вакансий Вашего направления

* Поля обязательные для заполнения

Просмотреть 🔍

Опубликовать ➔

Рисунок 3.12 – Створення резюме (етап 2)

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [12] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною(невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;

- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів(аж до порушення ДНК).

Відповідно до [13], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної(нагрів тканин і біологічних середовищ), електролітичної(розкладання крові і плазми) і біологічної(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [14]:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [15]: $Z_T / 3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де $R_\Pi = R_\phi + R_0$ - сумарний активний опір фазного R_ϕ і нульового R_0 дротів, Ом;

X_Π - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [18], має такий опір:

$$X_\Pi = 0,11 \text{ Ом};$$

$$R_\phi = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_\Pi = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (4.2) знаходимо повний опір петлі:

$$Z_\Pi = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (A)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (4.4.) :

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування (і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [16], СНіП, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [17] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [18] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні . За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції(з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устанавленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [19].

Стандарт [19] пред'являє жорсткі вимоги в таких областях: ергономіка(фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [20].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [21] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці (E_n) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розмішувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

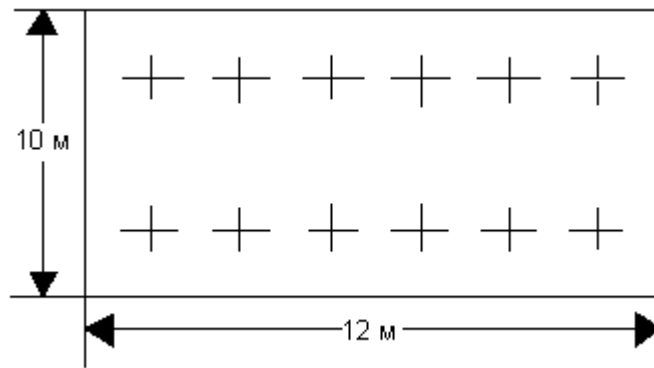


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 С ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 С, температура самозаймання 530 С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [22].

Згідно [23] приміщення відносяться до категорії В(пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [18] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту

з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.

- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- організація автоматичного контролю параметрів, що визначають джерела запалення;
- ліквідація можливості самозаймання речовин і матеріалів .
- Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :
- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

– робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [24] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [20]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ВИСНОВКИ

В ході дипломної роботи бакалавра була реалізована веб система електронної комерції, котра мала на меті полегшити задачу здобувачам та роботодавцям у пошуку співробітників. На прикладі даної роботи було розроблено інтерактивний інтернет-додаток, який поєднує у собі роботу з базою даних MySQL, практичну реалізацію механізму реєстрації, авторизації користувачів, основні принципи підписки на поштову розсилку. Також у роботі був розроблений і застосований алгоритм аутентифікації, який забезпечує надійний захист від злому паролів користувачів для додатків подібного рівня.

Взаємодія додатку з користувачем було виконано у вигляді простого та інтуїтивно-зрозумілого WEB-інтерфейсу.

В якості подальшого поліпшення і розширення функціональності проекту можна віднести додавання режиму попереднього модерування повідомлень, алгоритму захисту від прямого перебору паролів користувачів за допомогою обліку та блокування IP-адрес, можливість ведення логів, скриптів відображення розширеної статистики, модуля резервного копіювання даних.

При розробці питань охорони праці були виконані наступні завдання:

- проведений аналіз приміщення науково-дослідницької лабораторії з точки зору системи «Людина-Машина-Середовище»;
- виконали аналіз умов праці і характеру роботи;
- виділені основні небезпечні і шкідливі виробничі фактори, які діють в вказаному приміщенні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Schneider, G.T. «Electronic Commerce» [Текст] / Gary Schneider - Working Draft. Course Technology, 2008. – 185 с.
- 2) Makoto Y.B. «Electronic Commerce: Theory and Practice (Studies in Computational Intelligence)» [Текст] / Makoto Y.B., Springer, 2008. – 287 с.
- 3) Д.В. Котеров. Самовчитель PHP 5. [Текст] / Пітер 2011. – 570 с.
- 4) С. Суэринг, Т. Конверс, Д. Парк. PHP и MySQL [Текст] / пер. с англ./ Стив Суэринг, Тим Конверс, Джойс Парк - 2001. - 704 с..
- 5) К. Дари, Э. Баланеску. PHP и MySQL, Beginning PHP and MySQL E-Commerce: From Novice to Professional. [Текст] / К. Дари, Э. Баланеску, М.: «Вильямс», 2010. – 605 с.
- 6) Д. Котеров, А. Костарев; «PHP 5», серия «в подлиннике», [Текст] / Д. Котеров, А. Костарев, издательство BHV, 1120 с.
- 7) Мержевич, В.В. Російський сайт вивчення HTML та CSS2/3 технологій, В.В. Мержевич. [Електронний ресурс]. Режим доступа: [www/htmlbook.ru](http://www.htmlbook.ru)
- 8) Матраченко, Л.Б. - Sizzle как движок кроссбраузерных CSS-селекторов [Текст] / Л.Б. Матраченко 2010. - 377 с.
- 9) Gregory, L.D. HTML5 Differences from HTML 4 [Текст] / L.D. Gregory - Working Draft. World Wide Web Consortium 2011. - 134 с.
- 10) Gluschenko, T.M. W3C Working Draft [Текст] / T.M. Gluschenko - Working Draft. World Wide Web Consortium 2013. - 185 с.
- 11) Shawn McCool. Laravel Starter [Текст] / Shawn McCool - Packt Publishing (англ.) русск., 2012. - 64 с.
- 12) Державний стандарт України. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 13) Державний стандарт України. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

- 14) Державний стандарт України. ГОСТ 12.0.003-74 Опасные и вредные производственные факторы. Классификация.
- 15) Нормативно-правові акти з охорони праці. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів
- 16) Державний стандарт України. ГОСТ 12.1.009-76. ССБТ. Электробезопасность. Термины и определения.
- 17) Державні санітарні норми України. ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів.
- 18) Державні санітарні норми України. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
- 19) Державний стандарт України. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
- 20) TCO' 07 Certified Displays. © 2007 Copyright TCO Development AB
- 21) Державні санітарні норми і правила. ДСанПіН 3.3.2.007-98, Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
- 22) Державні будівельні норми ДБН В.2.5-28-2006. Природне і штучне освітлення
- 23) Державний стандарт України. ГОСТ 12.1.044-89 Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
- 24) Нормативні акти пожежної безпеки. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
- 25) Державний стандарт України. ГОСТ 12.1.004-91. "Система стандартов безопасности труда. Пожарная безопасность. Общие требования".
- 26) Нормативні акти пожежної безпеки. НАПБ А.01.001-2014 "Правила пожежної безпеки в Україні"

27) Нормативні акти пожежної безпеки. НАПБ Б.03.001-2004. Про затвердження Типових норм належності вогнегасників.

Додаток А. Лістинг коду

```

<?php

include ('install.php');

$link = mysql_connect($db_loc, $db_user, $db_pass);

if(!$link) echo "Не удалось подключиться к серверу";
else
{
    mysql_select_db($db_name);

    $opt = "CREATE TABLE OPTIONS
        ( id smallint(25) UNSIGNED NOT NULL AUTO_INCREMENT,
          Name VARCHAR(25) NOT NULL,
          value VARCHAR(25) NOT NULL,
          PRIMARY KEY (id)
        );";
    if (mysql_query($opt))
        echo "Создание таблицы завершено";
    else
        echo "Таблицу создать не удалось";

        $act = "CREATE TABLE ACTION
            (id smallint(25) NOT NULL AUTO_INCREMENT,
             action VARCHAR(25) NOT NULL,
             PRIMARY KEY (action)
            );";
    if (mysql_query($act))
        echo "Создание таблицы завершено";
    else
        echo "Таблицу создать не удалось";

        $sub = "CREATE TABLE SUBJECT
            (id int (5) UNSIGNED NOT NULL AUTO_INCREMENT,
             topic VARCHAR(25) NOT NULL,
             name VARCHAR(25) NOT NULL
             PRIMARY KEY (id)
            );";
    if (mysql_query($sub))
        echo "Создание таблицы завершено";
    else
        echo "Таблицу создать не удалось";

        $usr = "CREATE TABLE Users
            (id int UNSIGNED NOT NULL AUTO_INCREMENT index,
             login VARCHAR(25) NOT NULL,
             password VARCHAR(25) NOT NULL,
             contact VARCHAR(25) NOT NULL,
             access smallint(10) NOT NULL
            );";
    if (mysql_query($usr))
        echo "Создание таблицы завершено";
    else
        echo "Таблицу создать не удалось";

        $msg = "CREATE TABLE MESSAGES

```



```

        (id int NOT NULL AUTO_INCREMENT index,
        topic_id int index,
        user_id int index,
        action_id smallint,
        time VARCHAR(25) NOT NULL,
        time_live bigint,
        text VARCHAR(25) NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (topic_id) REFERENCES SUBJECT (id)
        FOREIGN KEY (action_id) REFERENCES action (id)
        );";
if (mysql_query($msg))
    echo "Создание таблицы завершено";
else
    echo "Таблицу создать не удалось";

    $mal = "CREATE TABLE MAIL
    (id int(25) NOT NULL,
    user_id int(25),
    email VARCHAR(25) NOT NULL,
    activation VARCHAR(25) NOT NULL,
    time bigint NOT NULL,
    PRIMARY key (id),
    FOREIGN KEY (user_id) REFERENCES users (id)
    );";
if (mysql_query($mal))
    echo "Создание таблицы завершено";
else
    echo "Таблицу создать не удалось";

    $mls = "CREATE TABLE MAILSUB
    (mail_id int(25) index,
    topic_id int(25) index,
    FOREIGN KEY (mail_id) REFERENCES MAIL (id),
    FOREIGN KEY (topic_id) REFERENCES SUBJECT (id)
    );";
if (mysql_query($mls))
    echo "Создание таблицы завершено";
else
    echo "Таблицу создать не удалось";

    $ses = "CREATE TABLE SESSIONS
    (user_id int(25),
    sid VARCHAR(25) NOT NULL,
    time bigint(25)
    FOREIGN KEY (user_id) REFERENCES users (id)
    );";
if (mysql_query($ses))
    echo "Создание таблицы завершено";
else
    echo "Таблицу создать не удалось";
}
class MysqlSearch
{
    function find($keywords)
    {
        # Create a keywords array
        $keywords_array = explode(" ", $keywords);

        # Select data query
        if (!$this->searchcolumns)
        {
            $this->searchcolumns = "*";
        }
    }
}

```

```

        $search_data_sql = "SELECT ".$this->searchcolumns." FROM
".$this->table;
    }
    else
    {
        $search_data_sql = "SELECT ".$this->
>entry_Identifier.", ".$this->searchcolumns." FROM ".$this->table;
    }

    # Run query, assigning ref
    $search_data_ref = mysql_query($search_data_sql);

    # Define $search_results_array, ready for population
    # with refined results
    $search_results_array = array();

    if($search_data_ref)
    {
        while($all_data_array = mysql_fetch_array($search_data_ref))
        {
            # Get an entry identifier
            $my_ident = $all_data_array[$this->entry_Identifier];

            # Cycle each value in the product entry
            foreach($all_data_array as $entry_key=>$entry_value)
            {
                # Cycle each keyword in the keywords_array
                foreach($keywords_array as $keyword)
                {
                    # If the keyword exists...
                    if($keyword)
                    {
                        # Check if the entry_value contains the
keyword

                        if(stristr($entry_value,$keyword))
                        {
                            $keywords_found_array[$keyword]++;
                        }
                    }
                    else
                    {
                        $keywords_found_array[$keyword]++;
                    }
                    unset($keyword);
                }
            }

            if(sizeof($keywords_found_array) ==
sizeof($keywords_array))
            {
                array_push($search_results_array,"$my_ident");
                break;
            }
            unset($keywords_found_array);
            unset($entry_key);
            unset($entry_value);
        }
    }

    $this->numresults = sizeof($search_results_array);
    # Return the results array

```

```

        return $search_results_array;
    }

    function setidentifier($entry_identifier)
    {
        $this->entry_identifier = $entry_identifier;
    }

    function settable($table)
    {
        $this->table = $table;
    }

    function setsearchcolumns($columns)
    {
        $this->searchcolumns = $columns;
    }
}

?>

<?php

    $query = "SELECT * FROM ".DB_CAT."";
    $result = $dbh->query($query);

    if(!$result) { throw new ExceptionMySQL($dbh->error,$query,"Error
executing SQL query!"); }

    if($result->num_rows>0)
    {
        while($row = $result->fetch_array())
        {
            echo          "<p><input          type=\"checkbox\"
value=\"\".$row['id_cat'].\"\" name=\"id_cat[]\">\".$row['name'].\"</p>";
        }
    }

    $url = "http://".$_SERVER["SERVER_NAME"].root()."form.php";

    $get_content = file($url);
    $get_content = implode($get_content, "\r\n");

    preg_match("/<div class=\"form\">(.*?)</div>/isU", $get_content,
$out);

    echo $out[1];

    catch(ExceptionObject $exc)
    {
        require_once("admin/lib/exception_object_debug.inc");
    }
    catch(ExceptionMySQL $exc)
    {
        require_once("admin/lib/exception_mysql_debug.inc");
    }
    catch(ExceptionMember $exc)
    {
        require_once("admin/lib/exception_member_debug.inc");
    }
    Error_Reporting(E_ALL & ~E_NOTICE);

    define("DEBUG", 0);

```

```

require "admin/class/class.exception_mysql.php";
require "admin/class/class.exception_object.php";
require "admin/class/class.exception_member.php";

try
{
    require_once "admin/lib/functions.inc";
    require_once "admin/lib/connect.inc";
}

$dbh->close();

catch(ExceptionObject $exc)
{
    require_once("admin/lib/exception_object_debug.inc");
}
catch(ExceptionMySQL $exc)
{
    require_once("admin/lib/exception_mysql_debug.inc");
}
catch(ExceptionMember $exc)
{
    require_once("admin/lib/exception_member_debug.inc");
}

?>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link rel="StyleSheet" type="text/css" href="admin/style.css">
<title><?php echo TITLE; ?></title>
</head>
<body>
<script type="text/javascript">

function ChoiceLang(Lang)
{
    document.forms[0].setAttribute('action','<?php
echo
$_SERVER['PHP_SELF']; ?>');
    document.forms[0].submit();
}

function validate_form()
{
    valid = true;

    if(trim(document.forms[1].dblocation.value) == "")
    {
        alert("<?php echo MSG_ALERT; ?> '<?php echo DBLOCATION;
?>'!");
        valid = false;
    }

    if(trim(document.forms[1].dblogin.value) == "")
    {
        alert("<?php echo MSG_ALERT; ?> '<?php echo DBLOGIN; ?>'!");
        valid = false;
    }

    if(trim(document.forms[1].dbname.value) == "")
    {
        alert("<?php echo MSG_ALERT; ?> '<?php echo DBNAME; ?>'!");
        valid = false;
    }
}

```

```

    }

    return valid;
}

function trim(str)
{
    var newstr = str.replace(/^\s*(.+?)\s*$/, "$1");

    if(newstr == " ")
    {
        return "";
    }

    return newstr;
}

</script>
<h1><?php echo TITLE_NAME; ?></h1>
<p>v <?php echo $version; ?></p>
<h2 style="text-align: center"><?php echo TITLE; ?></h2>
<br>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method=post>
    <table class="content" border="0">
        <tr>
            <td><?php echo FORM_LANGUAGE; ?>: </td>
            <td><select type="text" onchange='ChoiceLang(this)';
class="input" name="language">
                <option value="ru" <?php if($language == 'ru') echo
'selected'; ?><?php echo FORM_LANGUAGE_RU; ?></option>
                <option value="en" <?php if($language == 'en') echo
'selected'; ?><?php echo FORM_LANGUAGE_EN; ?></option>
            </select></td>
        </tr>
    </table>
</form>
<p>*-<?php echo REQUIREDFIELD; ?></p>
<div class="tableform">
    <form action='<?php echo $_SERVER['PHP_SELF']; ?>' onsubmit="return
validate_form();" method=post>
        <table border="0">
            <tr>
                <td><?php echo DBLOCATION; ?>*: </td>
                <td><input type="text" name="dblocation" value='<?php
if($_POST['dblocation']) echo $_POST['dblocation']; else echo "localhost";
?>'></td>
            </tr>
            <tr>
                <td><?php echo DBLOGIN; ?>*: </td>
                <td><input type="text" name="dblogin" value='<?php echo
$_POST['dblogin']; ?>'></td>
            </tr>
            <tr>
                <td><?php echo DBPASSWD; ?>: </td>
                <td><input type="text" name="dbpasswd" value='<?php echo
$_POST['dbpasswd']; ?>'></td>
            </tr>
            <tr>
                <td><?php echo DBNAME; ?>*: </td>
                <td><input type="text" name="dbname" value='<?php echo
$_POST['dbname']; ?>'></td>
            </tr>
            <tr>
                <td><?php echo PREFIX; ?>: </td>

```

```

        <td><input type="text" name="prefix" value='<?php
if($_POST['prefix']) echo $_POST['prefix']; else echo "sm_"; ?>'></td>
    </tr>
    <tr>
        <td><?php echo DBCHARSET; ?></td>
        <td><input type="text" name="dbcharset" value='<?php
if($_POST['dbcharset']) echo $_POST['dbcharset']; else echo "utf8"; ?>'></td>
    </tr>
    <tr>
        <td><input class="button" type="submit" value="<?php echo
FORM_CREATE; ?>"></td>
    </tr>
</table>
    <input type="hidden" name="language" value="<?php echo $language;
?>">
    </form>
</div>
<?php

if(!empty($_POST['dblocation']) &&
!empty($_POST['dblogin']) &&
!empty($_POST['dbname']))
{
    $_POST['dblocation'] = trim($_POST['dblocation']);
    $_POST['dblogin'] = trim($_POST['dblogin']);
    $_POST['dbpasswd'] = trim($_POST['dbpasswd']);
    $_POST['dbname'] = trim($_POST['dbname']);
    $_POST['prefix'] = trim($_POST['prefix']);
    $_POST['dbcharset'] = trim($_POST['dbcharset']);

    echo "<br><h2>".TITLE_NAME_2."</h2><ul>";

    $dbh = new mysqli($_POST['dblocation'], $_POST['dblogin'],
$_POST['dbpasswd'], $_POST['dbname']);

    if(mysqli_connect_errno())
    {
        echo "<li>".MSG_UNABLE_CONNECT_TO_DB."</li>";
        echo '</ul>';
    }
    else
    {
        echo "<li>".MSG_CONNECTION_ESTABLISHED."</li>";
        echo "<li>".MSG_DB_CREATED."</li>";

        $dbh->query("SET NAMES '". $_POST['dbcharset']."'");

        $query[] = "CREATE TABLE IF NOT EXISTS
'".$_POST['prefix']."aut` (
`passw` VARCHAR( 32 ) NOT NULL
) ENGINE=MyISAM";

        $query[] = "CREATE TABLE IF NOT EXISTS
'".$_POST['prefix']."attach` (
`id_attachment` int(7) NOT NULL AUTO_INCREMENT,
`name` VARCHAR( 255 ) NOT NULL,
`path` VARCHAR( 255 ) NOT NULL,
`id_send` int(7) NOT NULL,
PRIMARY KEY (id_attachment)
) ENGINE=MyISAM DEFAULT CHARSET=utf8";

        $query[] = "INSERT INTO `".$_POST['prefix']."aut` VALUES
('b59c67bf196a4758191e42f76670ceba)";

```

```

        $query[] = "CREATE TABLE IF NOT EXISTS
".$_POST['prefix']."category (
        id_cat int(7) NOT NULL auto_increment,
        name VARCHAR( 200 ) NOT NULL,
        PRIMARY KEY (id_cat)
        ) ENGINE=MyISAM DEFAULT CHARSET=utf8";

        $query[] = "INSERT INTO `".$_POST['prefix']."category` VALUES
(1, 'Category 1')";
        $query[] = "INSERT INTO `".$_POST['prefix']."category` VALUES
(2, 'Category 2')";
        $query[] = "INSERT INTO `".$_POST['prefix']."category` VALUES
(3, 'Category 3')";

        $query[] = "CREATE TABLE IF NOT EXISTS
".$_POST['prefix']."charset (
        `id_charset` int(5) NOT NULL AUTO_INCREMENT,
        `charset` VARCHAR( 32 ) NOT NULL,
        PRIMARY KEY (id_charset)
        ) ENGINE=MyISAM";

        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(1, 'utf-8')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(2, 'iso-8859-1')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(3, 'iso-8859-2')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(4, 'iso-8859-3')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(5, 'iso-8859-4')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(6, 'iso-8859-5')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(7, 'iso-8859-6')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(8, 'iso-8859-8')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(9, 'iso-8859-7')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(10, 'iso-8859-9')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(11, 'iso-8859-10')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(12, 'iso-8859-13')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(13, 'iso-8859-14')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(14, 'iso-8859-15')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(15, 'iso-8859-16')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(16, 'windows-1250')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(17, 'windows-1251')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(18, 'windows-1252')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(19, 'windows-1253')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(20, 'windows-1254')";
        $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(21, 'windows-1255')";

```

```

                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(22, 'windows-1256')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(23, 'windows-1257')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(24, 'windows-1258')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(25, 'gb2312')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(26, 'big5')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(27, 'iso-2022-jp')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(28, 'ks_c_5601-1987')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(29, 'euc-kr')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(30, 'windows-874')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(31, 'koi8-r')";
                $query[] = "INSERT INTO `".$_POST['prefix']."charset` VALUES
(32, 'koi8-u')";

                $query[] = "CREATE TABLE IF NOT EXISTS
`".$_POST['prefix']."log` (
        `id_log` int(7) NOT NULL AUTO_INCREMENT,
        `count` int(7) NOT NULL,
        `time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
        PRIMARY KEY (id_log)
    ) ENGINE=MyISAM";

                $query[] = "CREATE TABLE IF NOT EXISTS
`".$_POST['prefix']."ready_send` (
        `id_ready_send` int(10) NOT NULL AUTO_INCREMENT,
        `id_user` int(7) NOT NULL,
        `id_send` int(7) NOT NULL,
        PRIMARY KEY (`id_ready_send`),
        KEY `id_user` (`id_user`),
        KEY `id_send` (`id_send`)
    ) ENGINE=MyISAM";

                $query[] = "CREATE TABLE IF NOT EXISTS
`".$_POST['prefix']."send` (
        `id_send` int(7) NOT NULL AUTO_INCREMENT,
        `name` VARCHAR( 200 ) NOT NULL,
        `message` text NOT NULL,
        `prior` enum('1','2','3') NOT NULL DEFAULT '1',
        `date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
        `pos` int(7) NOT NULL,
        `id_cat` int(7) NOT NULL,
        `active` enum('yes','no') NOT NULL DEFAULT 'yes',
        PRIMARY KEY (id_send)
    ) ENGINE=MyISAM DEFAULT CHARSET=utf8";

                $query[] = "CREATE TABLE IF NOT EXISTS
`".$_POST['prefix']."settings` (
        `language` enum('en','ru') NOT NULL DEFAULT 'en',
        `count_send` int(4) NOT NULL DEFAULT '0',
        `count_user` int(4) NOT NULL DEFAULT '0',
        `email` VARCHAR( 200 ) NOT NULL,
        `organization` VARCHAR( 200 ) NOT NULL,
        `from_mail` VARCHAR( 200 ) NOT NULL,
        `smtp_host` VARCHAR( 200 ) NOT NULL,
        `smtp_username` VARCHAR( 200 ) NOT NULL,

```



```

`smtp_password` VARCHAR( 200 ) NOT NULL,
`smtp_port` int(8) NOT NULL DEFAULT '25',
`smtp_aut` enum('1','2') NOT NULL DEFAULT '1',
`smtp_ssl` enum('yes','no') NOT NULL DEFAULT 'no',
`send_server` enum('1','2') NOT NULL,
`id_charset` int(4) NOT NULL DEFAULT '0',
`ContentType` enum('1','2') NOT NULL DEFAULT '1',
`day` int(4) NOT NULL DEFAULT '0',
`count_interval` int(6) NOT NULL DEFAULT '1',
`send_limit` enum('yes','no') NOT NULL DEFAULT 'no',
`smtp_timeout` int(6) NOT NULL DEFAULT '5',
`limit_number` int(6) NOT NULL DEFAULT '100',
`many_send` enum('yes','no') NOT NULL DEFAULT 'no',
`del` enum('yes','no') NOT NULL DEFAULT 'yes',
`show_email` enum('yes','no') NOT NULL DEFAULT 'yes',
`newssubscribernotify` enum('yes','no') NOT NULL DEFAULT
'yes',
`reply` enum('yes','no') NOT NULL DEFAULT 'no',
`unsubscribe` enum('yes','no') NOT NULL,
`subjecttextconfirm` VARCHAR( 200 ) NOT NULL,
`textconfirmation` text NOT NULL,
`unsublink` text NOT NULL,
`interval_type` enum('no','m','h','d') NOT NULL DEFAULT 'no',
`precedence` enum('no','bulk','junk','list') NOT NULL DEFAULT
'bulk'
) ENGINE=MyISAM DEFAULT CHARSET=utf8";

$query[] = "INSERT INTO `".$_POST['prefix']."settings` VALUES
('".$language.", 5, 20, 'admin@mysite.com', '', '', 'smtp.gmail.com', '', '',
25, '1', 'no', '1', 1, '2', 7, 1, 'yes', 5, 100, 'yes', 'yes', 'yes', 'no',
'no', 'yes', 'Подписка на рассылку', 'Здравствуйте, %NAME%\r\n\r\nПолучение
рассылки возможно после завершения этапа активации подписки. У Вас %DAYS% дней,
чтобы активировать подписку, перейдите по следующей ссылке: %CONFIRM%\r\nЕсли
Вы не производили подписку на данный email, просто проигнорируйте это письмо
или перейдите по ссылке: %UNSUB%\r\nС уважением, администратор сайта
%SERVER_NAME%', 'Отписаться от рассылки: <a href=%UNSUB%>%UNSUB%</a>', 'no',
'bulk')";

$query[] = "CREATE TABLE IF NOT EXISTS
".$_POST['prefix']."subscription (
`id_sub` int(7) NOT NULL AUTO_INCREMENT,
`id_user` int(10) NOT NULL,
`id_cat` int(7) NOT NULL,
PRIMARY KEY (`id_sub`),
KEY `id_cat` (`id_cat`),
KEY `id_user` (`id_user`)
) ENGINE=MyISAM";

$query[] = "CREATE TABLE IF NOT EXISTS
".$_POST['prefix']."users` (
`id_user` int(10) NOT NULL AUTO_INCREMENT,
`name` VARCHAR( 200 ) NOT NULL,
`email` VARCHAR( 200 ) NOT NULL,
`ip` VARCHAR( 64 ) NOT NULL,
`cod` VARCHAR( 64 ) NOT NULL,
`time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`status` enum('active','noactive') NOT NULL DEFAULT
'noactive',
`time_send` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
PRIMARY KEY (id_user)
) ENGINE=MyISAM DEFAULT CHARSET=utf8";

$create_table = true;

```

```

        foreach($query as $value)
        {
            if(!@$dbh->query($value)) $create_table = false;
        }

        if($create_table == false)
        {
            echo
class=error>".MSG_FAILED_TO_CREATE_TABLES."
            ".$_POST['dbname'].!"
            ".MSG_THEY_ALREADY_EXIST."</span></li>";
        }
        else echo
            "<li>".MSG_TABLE_SUCCESSFULLY_CREATED."
            ".$_POST['dbname']."...</li>";

        $string='<?php

$db_location = "$_POST['dblocation']."; // database host
$db_name = "$_POST['dbname']."; // database name
$db_user = "$_POST['dblogin']."; // login
$db_passwd = "$_POST['dbpasswd']."; // password
$db_charset = "$_POST['dbcharset']."; // database charset

$version = "$_version.";

$dbh = new mysqli($db_location, $db_user, $db_passwd, $db_name);

if(mysqli_connect_errno())
{
    error("Error connecting to MySQL database: Database server
$db_location is not available!");
}

if(@file_exists("install.php") or @file_exists("../install.php"))
{
    error("Please remove install.php!");
}

$versql = $dbh->server_info;
list($major, $minor) = explode(".", $versql);
$ver = $major.".".$minor;

if((float)$ver >= 4.1 AND !empty($db_charset))
{
    $dbh->query("SET NAMES \"$_db_charset.\"");
}

define("DB_AUT", "$_POST['prefix'].'aut");
define("DB_ATTACH", "$_POST['prefix'].'attach");
define("DB_CAT", "$_POST['prefix'].'category");
define("DB_CHAR", "$_POST['prefix'].'charset");
define("DB_SETTING", "$_POST['prefix'].'settings");
define("DB_SUB", "$_POST['prefix'].'subscription");
define("DB_READY", "$_POST['prefix'].'ready_send");
define("DB_SEND", "$_POST['prefix'].'send");
define("DB_USERS", "$_POST['prefix'].'users");
define("DB_LOG", "$_POST['prefix'].'log");

define("CHARSET", "utf-8");

?>';

        if(@is_writable("admin/lib/connect.inc"))
        {
            $f = @fopen("admin/lib/connect.inc", "w");

```

```

        if(fwrite($f,$string) === FALSE)
        {
            echo "class=error>".MSG_ERROR_INSTALLATION_CANNOT_WRITE."</span></li></ul>";
            echo "class=error>".MSG_ERROR_INSTALLATION_CANNOT_WRITE_2."</span>";
        }
        else
        {
            echo "class=error>".MSG_INSTALLATION_IS_COMPLETE."</li></ul>";
            echo "class=error>".MSG_INSTALLATION_IS_COMPLETE_2."</span>";
        }
        fclose($f);
    }
    else
    {
        echo "class=error>".MSG_ERROR_INSTALLATION_FILE_ISNT_WRITABLE."</li></ul>";
        echo "class=error>".MSG_ERROR_INSTALLATION_FILE_ISNT_WRITABLE_2."</span>";
    }
}

?>
<p><?php echo LOGO; ?>, <?php echo AUTHOR; ?></p>
</body>
</html>

```

Додаток Б.
Електронні плакати

Дипломна робота

На тему:
«Веб система електронної біржі праці»

Керівник проекту:

Проф. Кривуля Г.Ф.

Здобувач вищої освіти:

Відчинкін О.А.

Сверодонецьк, 2018

Актуальність

- *Електронна комерція на сьогоднішній день це прискорення більшості бізнес-процесів за рахунок їх проведення електронним чином. Таким чином, електронну комерцію можна характеризувати як ведення бізнесу через Інтернет.*
- *Електронний бізнес сьогодні є:*
 - - *ефективним засобом проведення маркетингових досліджень та здійснення платежів;*
 - - *найкращою системою комунікацій, яка дає можливість встановити і підтримувати постійний зв'язок з будь-яким абонентом в світі (за умов його підключення до мережі);*
 - - *потужним джерелом економічної, фінансової та наукової інформації.*

Мета роботи

Метою дипломного проектування є створення і впровадження системи електронної комерції з працевлаштування з варіативним набором вакансій і резюме з метою полегшення роботи персоналу і виявлення відповідних форм вакансій і резюме претендентів.

3

Постановка задачі

- *Розробка системи працевлаштування*
- *Налагодження і усунення дефектів*
- *CRUD операції з вакансіями, резюме, категоріями*
- *Всілякий пошук за заданими критеріями*
- *Високі навантаження і її розподіл*

4

Застосовувані технології

- *HTML5, CSS3*
- *Framework Laravel*
- *JQuery*



Зовнішній вигляд: головна сторінка сайту з якою працює користувач



Зовнішній вигляд: фільтр за категоріями і регіонах.



Зовнішній вигляд: застосування фільтра

Регистрация
 Сонсатиль Работодатель
 * Обязательные поля

Е-mail *
 Пароль *
 Показать пароль

Ф.И.О. *
 Телефон *
 Получить новости на мой e-mail

Регистрация ➔

Регистрация
 Сонсатиль Работодатель
 * Обязательные поля

Е-mail *
 Пароль *
 Показать пароль

Название компании *
 Приемный работодатель Кандидат на работу

Отрасль *
 Город *
 Сайт

Контактная информация для менеджеров
 не отображать на сайте

Ф.И.О. *
 Телефон *
 Получить новости на мой e-mail

Регистрация ➔

Зовнішній вигляд: Форми реєстрації різних рівнів доступу

1 Выберите интересующие вас категории и города, в которых вы хотели бы получить на работу. Вы также можете указать ключевые слова (название должности). Количество выбранных категорий и городов не ограничено.

Категории *

<input type="checkbox"/> Информационные технологии	<input type="checkbox"/> IT-технологии/Интернет	<input type="checkbox"/> Банки/Финансы
<input type="checkbox"/> Бухгалтерия/Задат	<input type="checkbox"/> Психология/Психология детей	<input type="checkbox"/> Дипломатический персонал
<input type="checkbox"/> Жилищно-коммунальное хозяйство	<input type="checkbox"/> Инженеры/Технологи/Проектировщики	<input type="checkbox"/> Красота/Спорт/Тризм
<input type="checkbox"/> Политика/PR	<input type="checkbox"/> Маркетинг/Специальность по ЯЯ	<input type="checkbox"/> Медицина/Фармацевтика
<input type="checkbox"/> Недвижимость	<input type="checkbox"/> Образование/Наука	<input type="checkbox"/> Охрана безопасности
<input type="checkbox"/> Перевозчики	<input type="checkbox"/> Полиграфия	<input type="checkbox"/> Производство
<input type="checkbox"/> Работа для пенсионеров / людей с ограниченными возможностями	<input type="checkbox"/> Работа для студентов	<input type="checkbox"/> Работа за рубежом
<input type="checkbox"/> Работники/Служащие	<input type="checkbox"/> Работа на дому	<input type="checkbox"/> Работа на выезде
<input type="checkbox"/> Секретари/Офис-менеджеры	<input type="checkbox"/> Работа	<input type="checkbox"/> Рыночные
<input type="checkbox"/> Служба	<input type="checkbox"/> Услуги клиентов	<input type="checkbox"/> Сетевой маркетинг/Работа в Интернет
<input type="checkbox"/> Среды обслуживания	<input type="checkbox"/> Сторонние	<input type="checkbox"/> Стратегическое
<input type="checkbox"/> Торговля/Розница	<input type="checkbox"/> Телекоммуникации/Связь	<input type="checkbox"/> УДО/Менеджмент
<input type="checkbox"/> Юридические	<input type="checkbox"/> Транспорт/Автомотив	<input type="checkbox"/> Школа/Бизнес

Города *

<input type="checkbox"/> Вся Украина	<input type="checkbox"/> Александрия	<input type="checkbox"/> Алушты
<input type="checkbox"/> Вилка (Донецк)	<input type="checkbox"/> Бердянск	<input type="checkbox"/> Волынская
<input type="checkbox"/> Винница	<input type="checkbox"/> Голотовка	<input type="checkbox"/> Днепропетровский
<input type="checkbox"/> Днепропетровский	<input type="checkbox"/> Днепр	<input type="checkbox"/> Дрогобыч
<input type="checkbox"/> Донецк	<input type="checkbox"/> Житомир	<input type="checkbox"/> Ивано-Франковск
<input type="checkbox"/> Закарпатье	<input type="checkbox"/> Иван-Франковск	<input type="checkbox"/> Киев
<input type="checkbox"/> Киев/Подольский	<input type="checkbox"/> Киев	<input type="checkbox"/> Киев
<input type="checkbox"/> Кировоград	<input type="checkbox"/> Кременчук	<input type="checkbox"/> Кривой Рог
<input type="checkbox"/> Кривой Рог	<input type="checkbox"/> Крым	<input type="checkbox"/> Луганск
<input type="checkbox"/> Луцк	<input type="checkbox"/> Львов	<input type="checkbox"/> Мариуполь
<input type="checkbox"/> Мариуполь	<input type="checkbox"/> Москва	<input type="checkbox"/> Николаев
<input type="checkbox"/> Миколаїв	<input type="checkbox"/> Новомиловское	<input type="checkbox"/> Одесса
<input type="checkbox"/> Одесса	<input type="checkbox"/> Одесса	<input type="checkbox"/> Ровно
<input type="checkbox"/> Полтава	<input type="checkbox"/> Полтава	<input type="checkbox"/> Сєвєродонецький
<input type="checkbox"/> Сєвєродонецький	<input type="checkbox"/> Тернополь	<input type="checkbox"/> Ужгород
<input type="checkbox"/> Сумы	<input type="checkbox"/> Харьков	<input type="checkbox"/> Харьков
<input type="checkbox"/> Харьков	<input type="checkbox"/> Харьков	<input type="checkbox"/> Харьков
<input type="checkbox"/> Черновики	<input type="checkbox"/> Черкассы	<input type="checkbox"/> Чернигов
<input type="checkbox"/> Черкассы	<input type="checkbox"/> Чернигов	

2 Нажмите кнопку "Направить"

Направить ➔

Зовнішній вигляд: новинна розсилка

Зовнішній вигляд: Розміщення вакансії

11

Внешний вид: размещение резюме

12

Охорона праці

При розробці питань охорони праці були виконані наступні завдання:

- Проведено аналіз приміщення науково-дослідної лабораторії з точки зору системи «людина-машина-Середовище»;
- Виконали аналіз умов праці та характеру роботи;
- Виділено основні небезпечні і шкідливі виробничі фактори, які діють в зазначеному приміщенні.

23

Висновки

У процесі виконання дипломного проекту була реалізована веб система працевлаштування метою якої, полегшити задачу здобувачам і роботодавцям у пошуку співробітників. Система є простою і зручною у використанні. Дотримані вимоги до функціонала програми.

24