МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Web-технології Ч.2»

на тему:

**«Портал для пошуку роботи»**

студента ІI курсу групи ІПЗ-20-3

спеціальності 121 «Інженерія програмного забезпечення»

Башманівського Максима Олексійовича\_

(прізвище, ім’я та по-батькові)

Керівник ктн, доц. А.В. Морозов ,

Дата захисту: " 19 " січня 2022 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Граф \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Г. Чижмотря. .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Кузьменко \_

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ А.В.Морозов

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

\_Новічкову Єгору Максимовичу\_

1. Тема роботи: Портал для пошуку роботи,

керівник роботи: к.т.н., доцент А.В. Морозов.

1. Строк подання студентом: “ 12 ” січня 2021р.
2. Вхідні дані до роботи: Розробити інтернет-портал для пошуку роботи.
3. Зміст розрахунково-пояснювальної записки:
   * 1. Постановка завдання
     2. Аналіз аналогічних розробок
     3. Алгоритми роботи програми
     4. Опис роботи програми
     5. Програмне дослідження
4. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)
   * + 1. Презентація до КП
       2. Посилання на репозиторій: <https://gitlab.com/ipz203_bmo/kurs_project_22>
5. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1,2,3 | О.Г. Чижмотря, старший викладач кафедри інженерії програмного забезпечення |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 15 ” вересня 2021 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсового проекту | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі | 18.09-04.10 | виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 04.10-22.10 | виконано |
| 3 | Формулювання технічного завдання | 23.10-29.10 | виконано |
| 4 | Опрацювання літературних джерел | 30.10-17.11 | виконано |
| 5 | Проектування структури | 18.11-29.11 | виконано |
| 6 | Написання програмного коду | 30.11-02.01 | виконано |
| 7 | Відлагодження | 03.01-13.01 | виконано |
| 8 | Написання пояснювальної записки | 13.01-18.01 | виконано |
| 9 | Захист | 19.01 | виконано |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Башманівський М.О.

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Морозов

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Моїм завданням на курсову роботу була розробка інтернет – порталу для пошуку роботи.

Пояснювальна записка до курсової роботи на тему «Портал для пошуку роботи» складається з: вступу, основної частини (трьох розділів), висновків, списку використаної літератури та додатків.

Текстову частину трактовано на 28 сторінках друкованого тексту. Пояснювальна записка має 5 сторінок додатків. Список використаних джерел містить 6 найменувань і займає 0 сторінку. В роботі наведено 17 рисунків. Загальний обсяг роботи – 33 сторінок.

Ключові слова: ІНТЕРНЕТ ПОРТАЛ, ВЕБ-ДОДАТОК, MVS, БАЗА ДАНИХ, PHP, HTML, CSS, JS.

Зміст

ВСТУП6

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ7

1.1 Аналіз задачі, засобів та методів її вирішення7

1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту9

Висновки до розділу14

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ15

2.1 Проектування загального алгоритму роботи програми15

2.2 Розробка функціональних алгоритмів роботи програми16

2.3 Розробка програмного забезпечення20

Висновки до розділу22

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ23

3.1 Опис роботи з програмним додатком23

3.2 Тестування роботи програмного забезпечення26

Висновки до розділу28

ВИСНОВКИ29

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ28

ДОДАТКИ29

**ВСТУП**

**Актуальність теми.**

З переходом в еру технологій актуальність пошуку роботи не знижується, якщо порівнювати с минулими десятиліттями, навпаки, стає все відповідальнішим етапом в житті кожної людини. Для пошуку роботи люди використовують найрізноманітніші джерела інформації. Це може бути вакансія роздрукована в газеті, інформація від знайомих або заява до центру зайнятості. Однак в наш час, з появою інтернету, з’явився новий спосіб отримання інформації що до пошуку роботи. Зараз створюються спеціальні інтернет – портали, на яких ви маєте змогу розмістити резюме або відгукнутися на вакансію. Перевагою таких сайтів є те що вони мають структурований вид інформації, у порталі присутня пошукова система. Люди мають можливість розглянути вакансії, які задовольняють саме їх потреби. Інформація на таких сайтах завжди актуальна, тому пошук роботи не займе багато часу. Таким чином, на сьогодні, більшість використовує саме такі веб – додатки.

**Об’єктом дослідження** даноїкурсоої роботи єстворення сайту для пошуку роботи.

**Предметом дослідження** є реалізація функцій для менеджменту інформації у базі даних MySQL за допомогою мови програмування PHP, створення графічного інтерфейсу, за допомогою мов програмування HTML, CSS та JavaScript.

**Метою дослідження** єреалізація та засвоєння знань отриманих при вивченні дисципліни «Web-технології Ч.2»

**РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ**

**1.1 Аналіз задачі, засобів та методів її вирішення**

Задачею курсового проекту є створення порталу для пошуку роботи. Це завдання передбачає собою створення веб - додатку, що має графічний інтерфейс та виконує такий набір функцій, як: публікація новин, публікація вакансій та резюме, можливість редагування опублікованої інформації, створення власного кабінету, в якому буде зберігатись певна інформація про користувача.

Отже основними задачами при виконанні проекту будуть:

* Графічний інтерфейс;
* Реалізація функцій редагування опублікованої інформації;
* Реалізація відображення інформації, що зберігається на сервері.
* Реалізація функцій публікації інформації

Сайт повинен мати інтуїтивно зрозумілий графічний інтерфейс. Елементи керування порталом мають бути привабливі на вид, мати певний дизайн. Наявність певної системи адміністрування, яка дозволить виконувати контроль що до публікованої інформації. Також ця система повинна надавати можливість редагування основної інформації на сайті. Адміністрування не має бути доступним звичайним користувачам.

Для реалізації поставлених цілей потрібно обрати певне крос – платформове інтегроване середовище розробки. На даний час на вибір доступні такі додатки як: «PhpStorm» від JetBrains та «WebStorm» від JetBrains. При виборі я відштовхувався від знання графічного інтерфейсу середовища та наявності певних елементів, які допоможуть мені в реалізації. У проекті будуть використані такі мови програмування як: “PHP” для реалізації серверної частини, “JavaScript” для реалізації керування графічних елементів, “CSS”, “HTML” для розмітки та надання певного дизайну сайту. Невід’ємною частиною сайту є використання бази даних. Я обрав саме MySQL як систему керування базами даних, так як маю досвід роботи з нею. Для налаштування системи буде використано веб – додаток “PHPMyAdmin”. Сайт буде працювати на сервері з програмним забезпеченням Apache так як це веб – сервер який підтримує безліч можливостей, серед них – наявність інтерфейсу для підтримки мови програмування “PHP”.

**1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту**

Переглядаючи вже існуючі веб – додатки для пошуку роботи стало зрозуміло що існує безліч дуже схожих порталів для пошуку роботи. В даному розділі я розгляну найпопулярніші з тих що доступні для мене.

1. “Robota.ua” – українська компанія, яка розвиває інтернет – рекрутмент з 2000 року.

Головним активом компанії є сайт robota.ua – портал для пошуку роботи. Щомісячна відвідуваність перевищує чотири млн. унікальних відвідувачів, а загальна складає понад 10 млн. Використовуючи передові сервіси у сфері рекрутменту, понад 140 тисяч компаній розміщують свої вакансії й успішно знаходять цільових кандидатів.

Сайт має приємний дизайн, наявний весь можливий функціонал що притаманний сайтам даного типу, а саме: створення акаунту; розміщення вакансій та резюме; перегляд існуючих публікацій; редагування вже розміщеної інформації, зворотній зв’язок з роботодавця та шукача роботи.

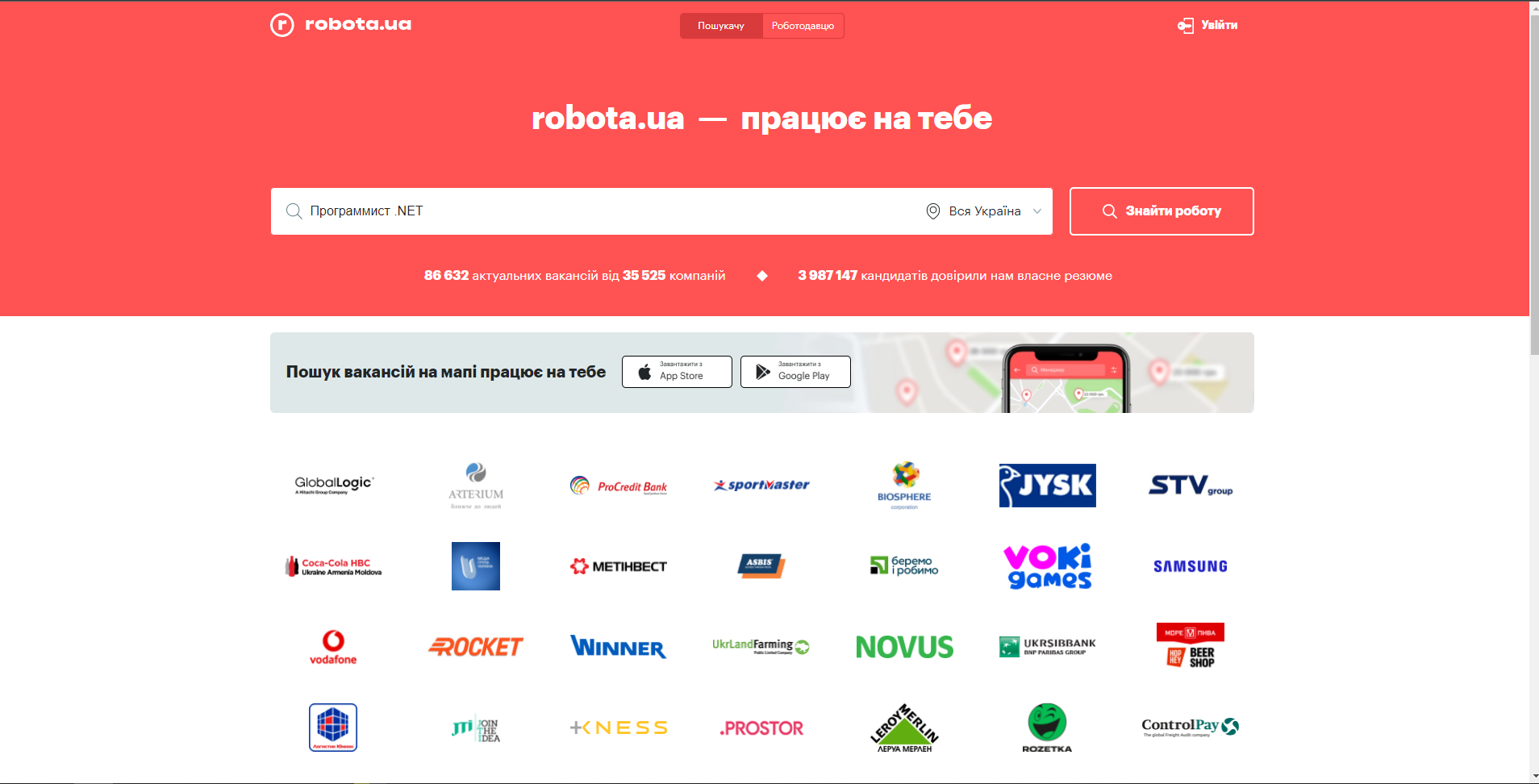


Рис. 1.1 Головне меню сайту для шукачів роботи Robota.ua

На рисунку 1.1 видно що сайт поділений на дві частини: контент для шукачів роботи та роботодавців. Для шукача наявна інформація про компанії що надають вакансії та популярні професії та міста, в яких отримують роботу. Також наявний пошук вакансій.

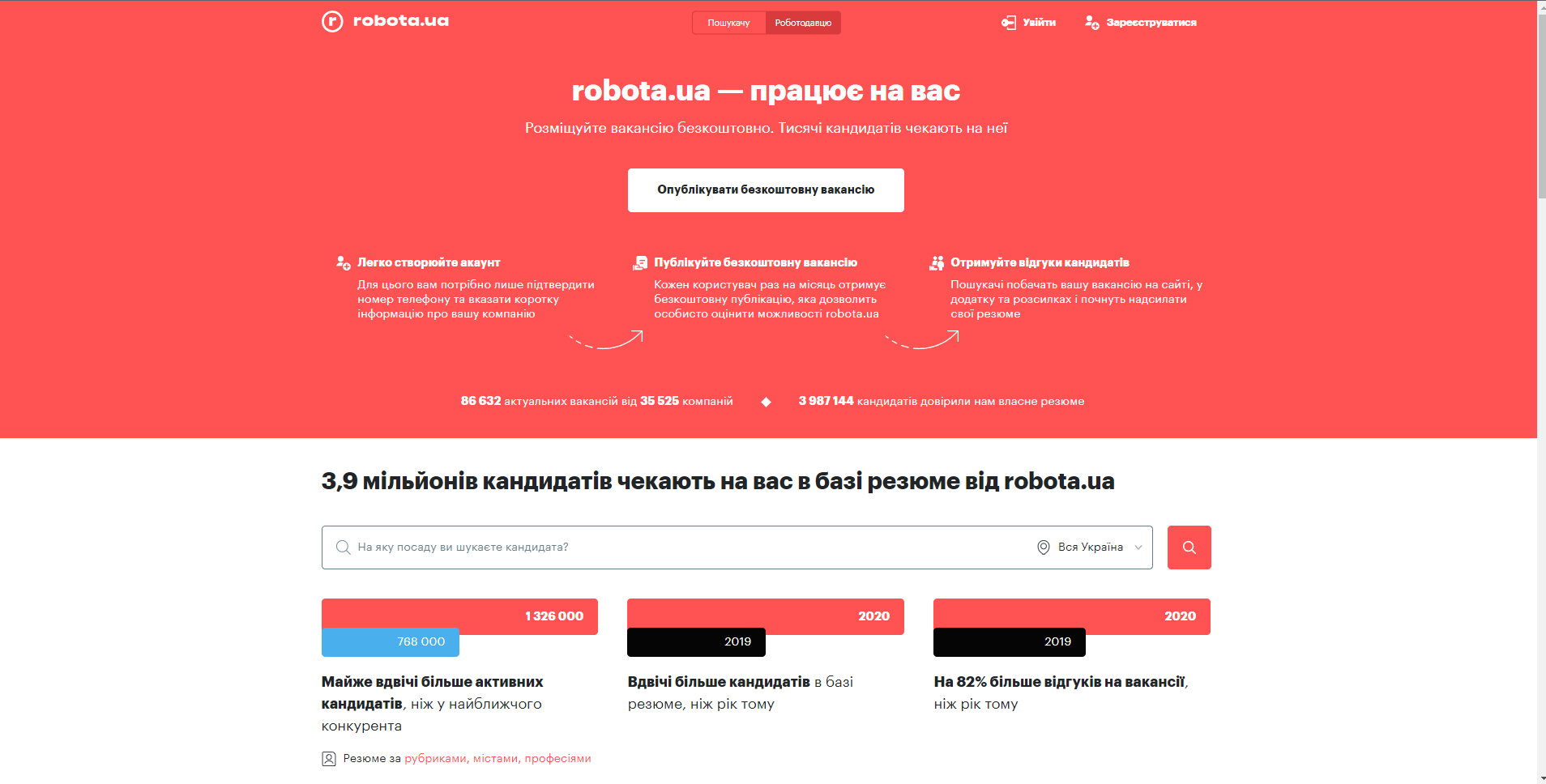


Рис. 1.2 Головне меню сайту для роботодавців Robota.ua

На рисунку 1.2 вже відображена інформація для роботодавців, наявна певна статистика що до знайдених кандидатів, є поле для пошуку резюме

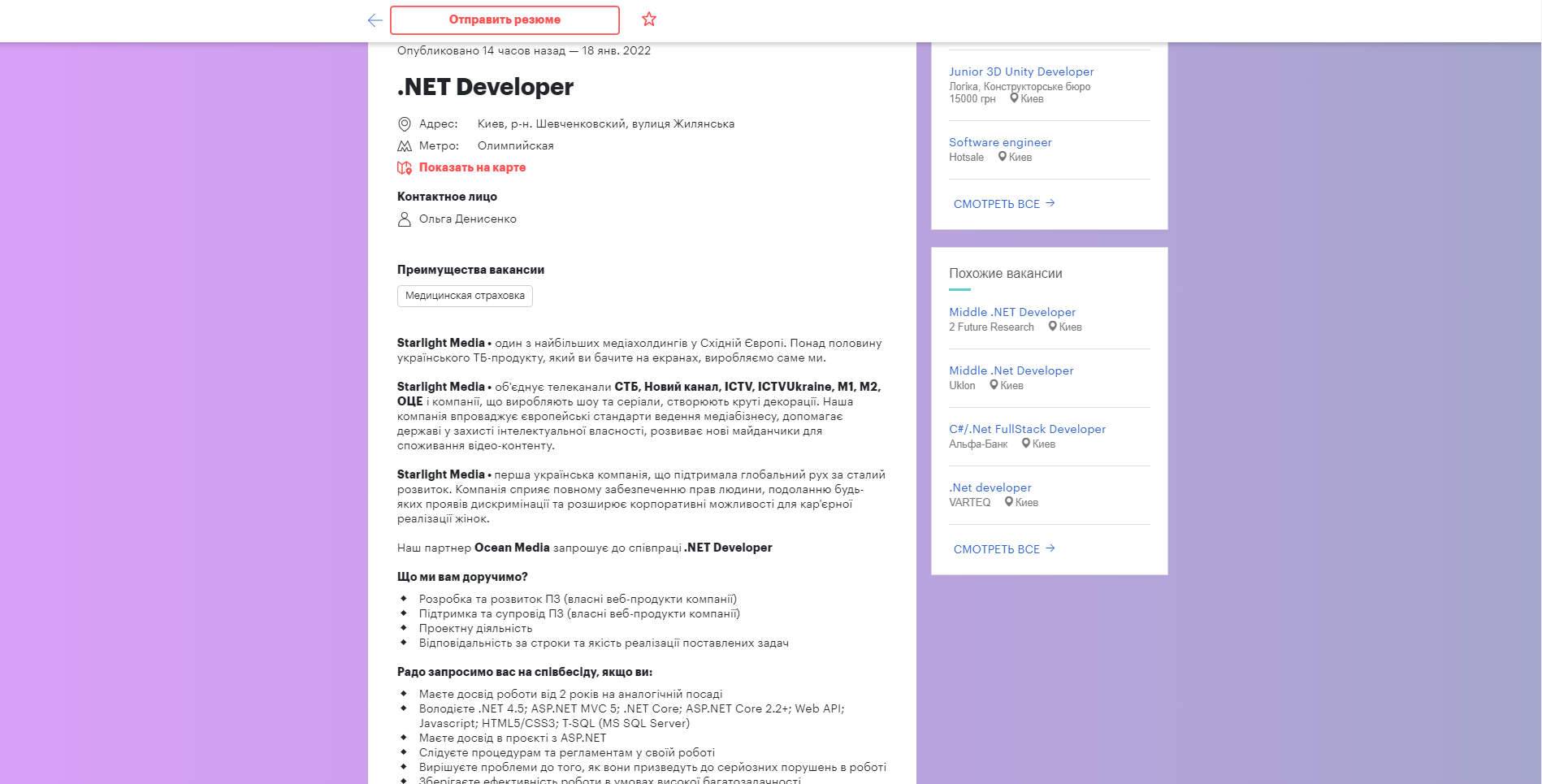


Рис. 1.3 Детальна інформація про вакансію Robota.ua

На наступному рисунку 1.3 видно детальне відображення інформації про вакансію, вказані обов’язки та переваги даної роботи, наявні контакти, з допомогою яких шукач роботи може зв’язатись з компанією. Також є функція відправки власного резюме. При пошуку інформації на сайті наявні всі можливі фільтри, з допомогою яких можна обрати вакансії, які доступні в місті в якому ви проживаєте. На сайті є можливість відправки на пошту користувача щойно доданих вакансій

До мінусів даного додатку можна віднести можливість оформлення платної підписки з додатковими можливостями. Також, сайт повільно працює, при будь - якій дії відбувається перезавантаження сторінки.

1. “WORK.UA” - український сайт з пошуку роботи, його щомісяця відвідують понад 14 млн людей, 155 тис. — розміщують вакансії. За даними на серпень 2017 року на сайті зібрано близько 2 млн резюме та 90 тисяч вакансій. Сайт пошуку роботи Work.ua створено дніпровською компанією «Реактор» 2006 року.

Сайт має звичний дизайн, на сайті є майже всі функції для коректного перегляду та розміщення інформації. На сайті також наявна функція розсилки вакансій на пошту.

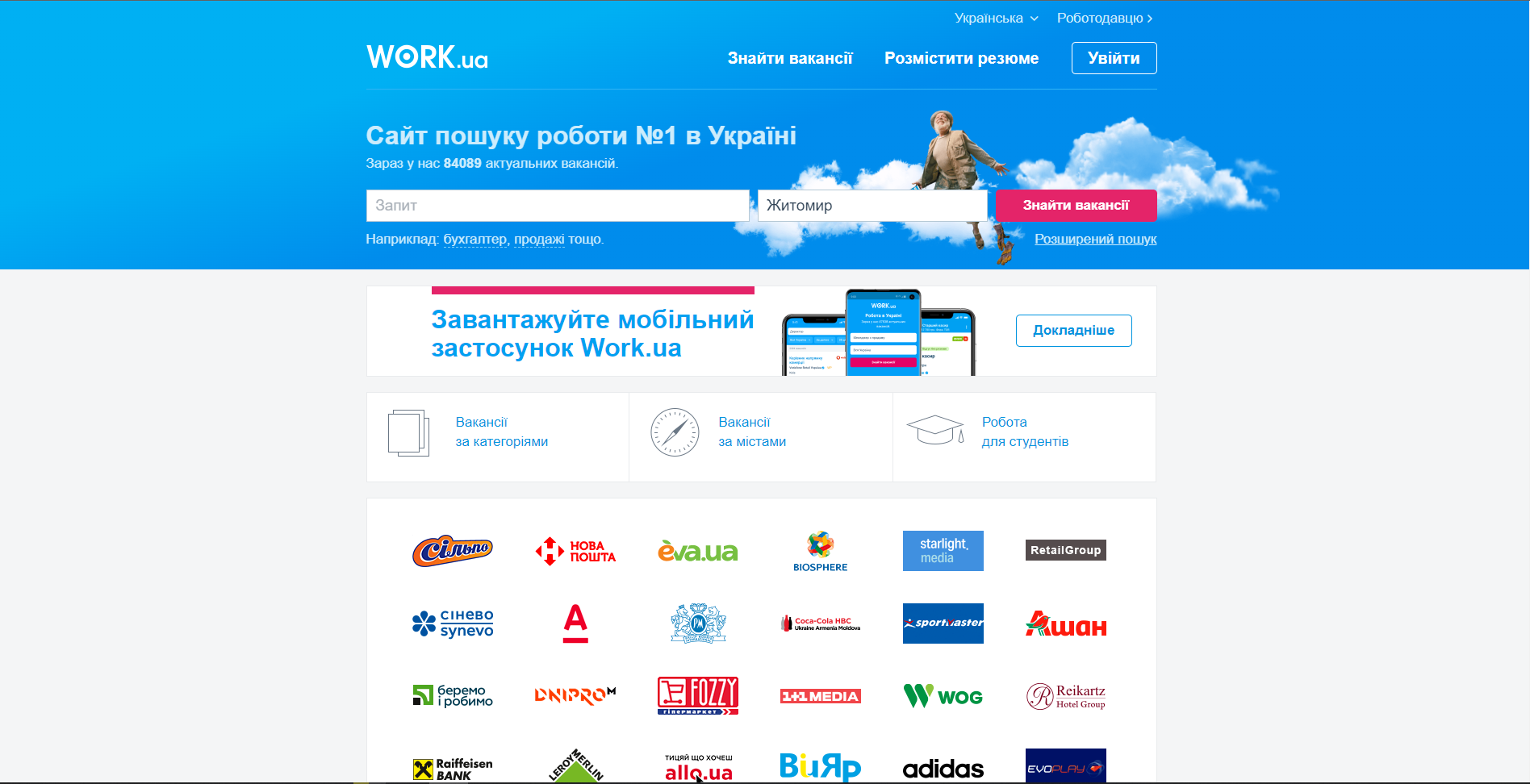


Рис. 1.4 Головне меню сайту WORK.ua

На рисунку 1.4 видно що портал для пошуку роботі розділений: для шукача роботи, для роботодавця. На головній сторінці розміщені компанії що надають вакансії для сайту, панель для пошуку розміщена у хедері сайту. Також видно що наявний розділ новинами та статтями, є допомога користувачам у створенні вакансії, резюме.

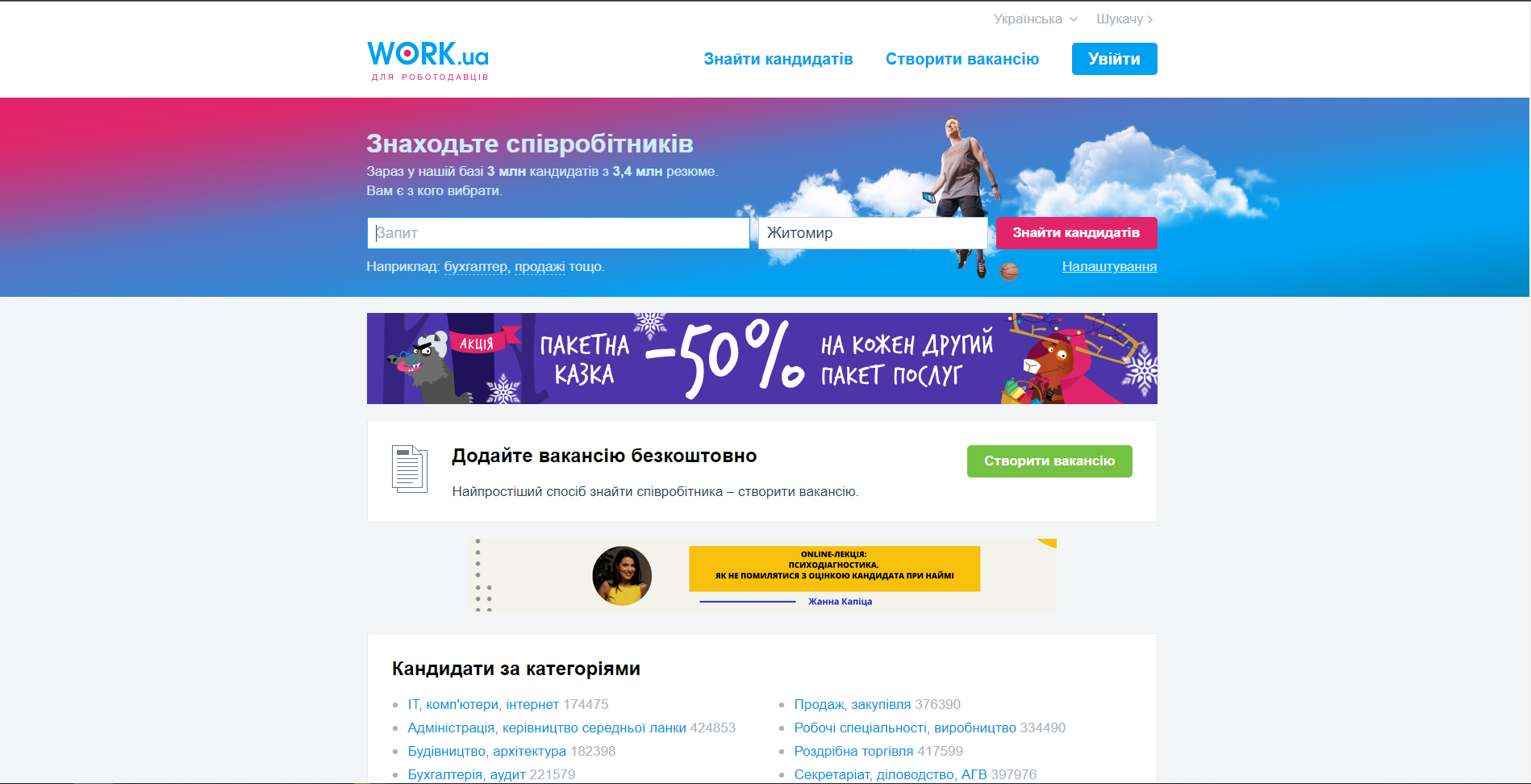


Рис. 1.5 Головне меню сайту WORK.ua, сторінка для роботодавців

На рисунку 1.5 видно наскільки відрізняється дизайн сторінки роботодавця і шукача. Одразу після хедера наявна реклама, яка займає багато місця. Далі розміщені фільтри для пошуку резюме. Наявна невелика статистика що до середньої зарплати в Україні

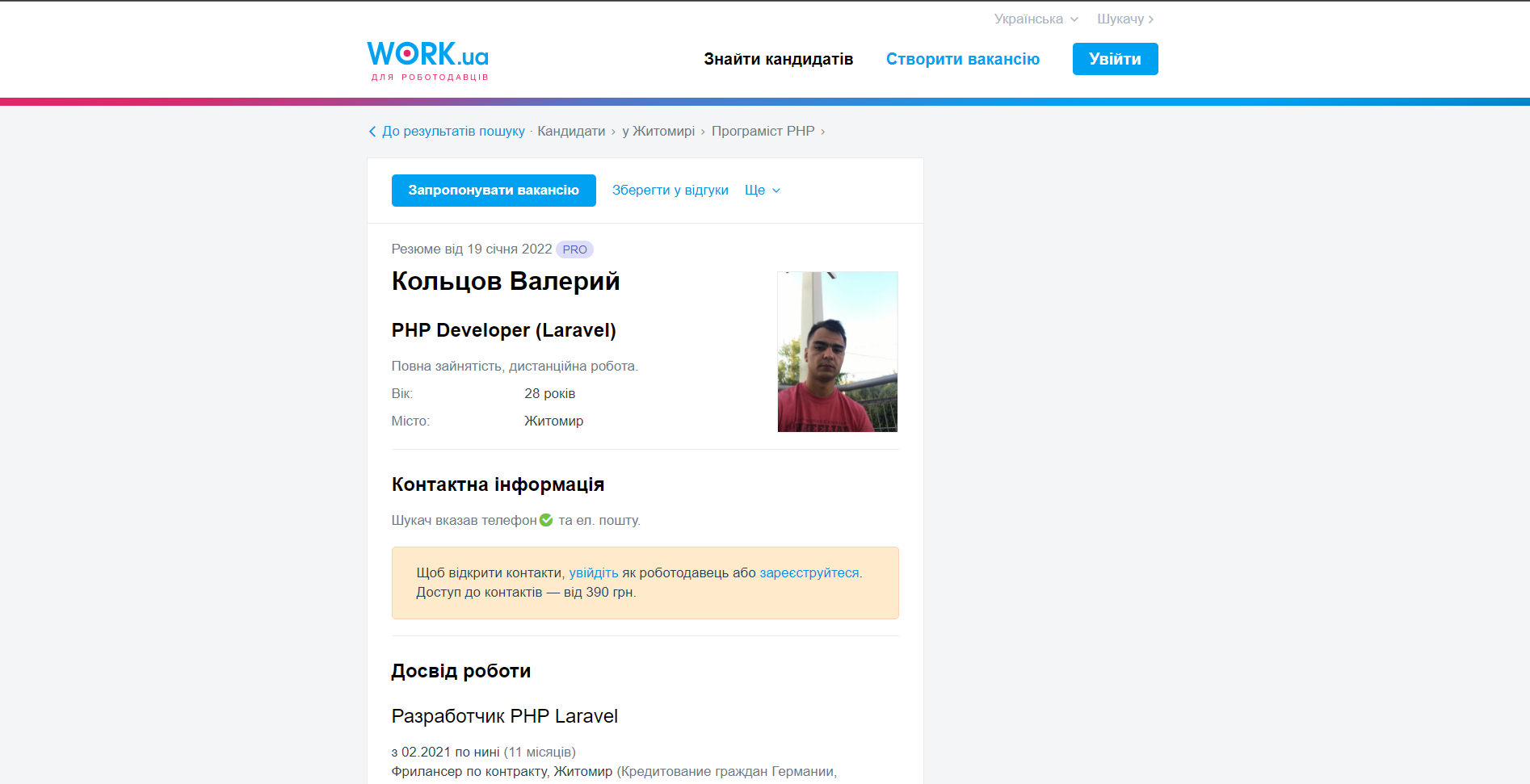


Рис. 1.6 сторінка перегляду резюме WORK.ua

На даному рисунку 1.6 видно дизайн сторінки перегляду резюме, вказана контактна інформація, досвід роботи та навички. До плюсів даного сайту можна віднести швидкодію, компактність публікованої інформації. До мінусів я віднесу неприємний дизайн на сторінці для роботодавців, наявність реклами.

Проаналізувавши дані сайти можна створити таку таблицю:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Зручність** | **Дизайн** | **Насиченість функціями** |
| **Robota.ua** | 10 | 10 | 9 |
| **WORK.ua** | 9 | 8 | 8 |

*Табл. 1.1 Порівняння аналогів*

**Висновки до першого розділу**

В першому розділі відбувся аналіз задачі, засобів та методів її вирішення. Виконано аналіз небагатьох існуючих сайтів з тематикою моєї курсової роботи, та їх порівняння. Були обрані засоби для створення програмного забезпечення за темою роботи, та обрано функціонал, який буде реалізовано. Для написання коду буде використано крос-платформове інтегроване середовище розробки для мови програмування PHP – PHPstorm. Створення та налаштування серверу буде відбуватись за допомогою програмног забезпеченням Apache. Візуалізація даних буде програмуватись з допомогою: HTML, CSS, JavaScript. Зберігатись дані будуть у базі даних яка працює як система керування базами даних MySQL.

Проаналізувавши ринок можливо зробити висновок що вже існують повноцінні веб – додатки по пошуку роботи, які надають насичений функціонал користувачам.

При розробці сайту будуть враховані деякі особливості та недоліки цих сайтів. Створення базового функціоналу цих сайтів є найголовніше.

**РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**2.1 Проектування загального алгоритму роботи програми**

Проаналізувавши отриману інформацію при дослідженні аналогів, було виведено перелік функцій, що потрібно буде реалізувати в процесі роботи для повноцінного існування сайту:

* Створення та редагуавання акаунту користувачів
* Функціонал для адміністратора
* Реалізація можливості публікування, редагування та видалення вакансій та резюме користувачами,
* Реалізація функцій перегляду вакансій та резюме
* Система пагінації для контенту сайту

Доступність функцій буде визначатись на етапі аутинтифікації користувача. Адміністратор буде мати змогу редагувати головну сторінку, видаляти та редагувати резюме та вакансії. Простий користувач матиме змогу публікації вакансій та резюме, перегляд опублікованого матеріалу.

На головній сторінці користувач буде бачити новини та панель з допомогою у створенні резюме та вакансій. Не аутинтифікований користувач матиме змогу перегляду матеріалу без можливості публікації.

**2.2 Розробка функціональних алгоритмів роботи програми**

Сформувавши загальний алгоритм роботи сайту було вирішено розробити вже існуючий патерн програмування MVC (Model-view-controller).

Це архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача. Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

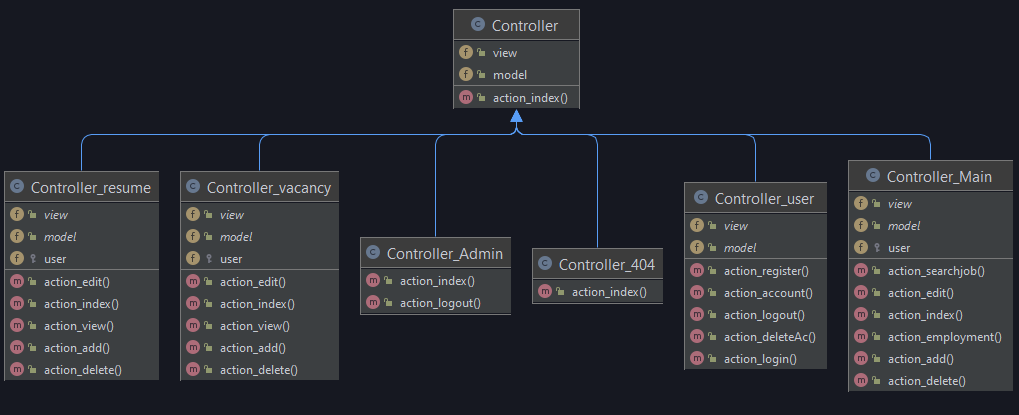


Рис. 2.1 система класів Controller

На рисунку 2.1 зображена система класів Controller. Це компонент, що зв'язує логіку аплікації (модель) із виглядом. Всього у моєму проекті було використано 6 контролерів:

* Controller\_resume (дії пов’язані з резюме)
* Controller\_vacancy (дії пов’язані з вакансіями)
* Controller\_Admin (дії адміністратора, можливості)
* Controller\_404 (Виклик неіснуючої сторінки при помилковій маршрутизації)
* Controller\_user (дії користувача)
* Controller\_main (головна сторінка, можливості її редагування адміністратором)

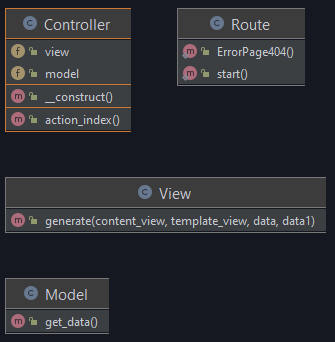


Рис. 2.2 система класів Core

На рисунку 2.2 зображена система класів Core. Це компонент, що є батьком для всіх інших систем. Нащадки цих класів будуть перевизначатись. Також тут наявний важливий елемент, який відповідає за маршрутизацію на сайті – Route, він приймає та створює контролер, викликає подію – action.

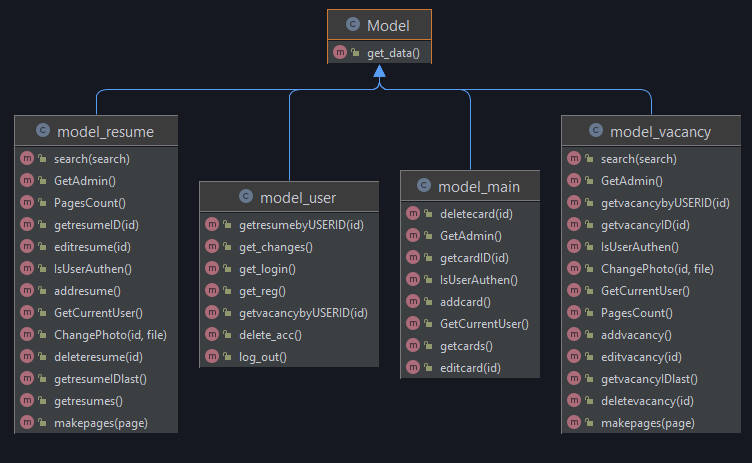


Рис. 2.3 система класів Model

На рисунку 2.3 зображена система класів Model. Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку. Тут здійснюються всі звернення до бази даних.

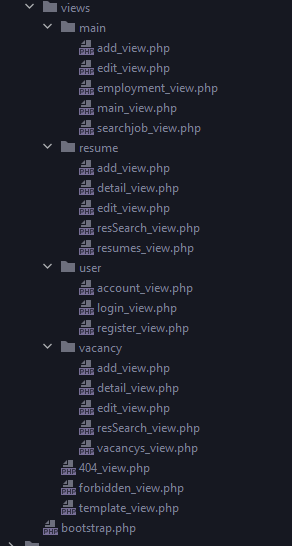


Рис. 2.3 структура папки views

На рисунку 2.3 представлена структура аплікацій які будуть відображатись з допомогою контролера.

При переході на сайт відбувається зчитування URL-адреси(маршруту) та визначення контролеру на основі якого будуть виконані певні дії, функції реалізовані в контролері. До кожного елементу сайту є свій контролер, який містить багато методів, які в свою чергу звертаються до моделі, де отримують відповідні аплікації та дані з бази даних.

**2.3 Розробка програмного забезпечення**

При розробці сайту було створено систему класів та їх методи:

За головне меню відповідають класи Controler\_Main, Model\_main та 5 сторінок аплікацій. Контролер має такі функції: index() – відображення новин та інших елементів, add() – додавання новин, edit() – редагування новин, delete – видалення новини.

За систему користувачів відповідають класи Controler\_user, Model\_user та 3 сторінки аплікацій. Контролер має такі функції: add() – додавання користувачів , edit() – редагування інформації користувачів, delete – видалення користувача, logout() – вихід користувача. У моделі відбувається звернення до бази даних, відповідної таблиці users

За створення вакансій відповідають класи Controler\_vacancy, Model\_vacancy та 5 сторінок аплікацій. Контролер має такі функції: index() – відображення вакансій, add() – додавання вакансії користувачем, edit() – редагування вакансії користувачем, delete – видалення вакансії. У моделі відбувається звернення до бази даних, відповідної таблиці vacancy.

За створення резюме відповідають класи Controler\_resume, Model\_resume та 5 сторінок аплікацій. Контролер має такі функції: index() – відображення резюме, add() – додавання резюме користувачем, edit() – редагування резюме користувачем, delete – видалення резюме. У моделі відбувається звернення до бази даних, відповідної таблиці resume.

У роботі використано систему пагінації на таких сторінках як:

* Resume
* Vacancy

Майже всі поля на сайті проходять валідацію та перевірку.

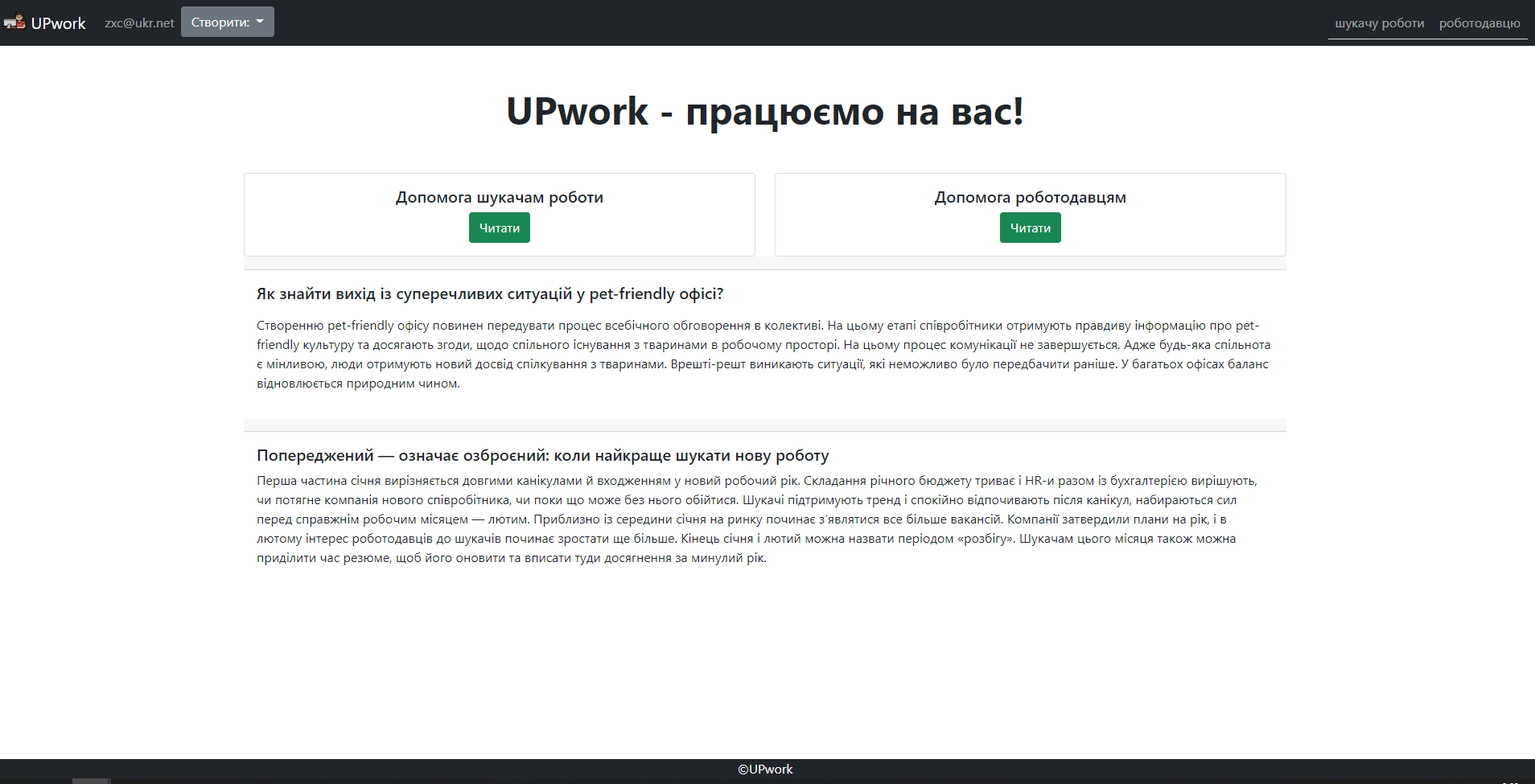
Для коректного відображення контенту було використано клієнтський фреймворк bootstrap 5. Були задіяні картинки, для використання їх користувачами та для дизайну сайту. Всі файли зберігаються в папці images. Код JavaScript в папці js, деякі стилі в папці css.



Рис. 2.4 структура створеної бази даних



Рис. 2.5 головне меню адміністратора



*Рис. 2.6 Вигляд сайту користувача*

**Висновки до другого розділу**

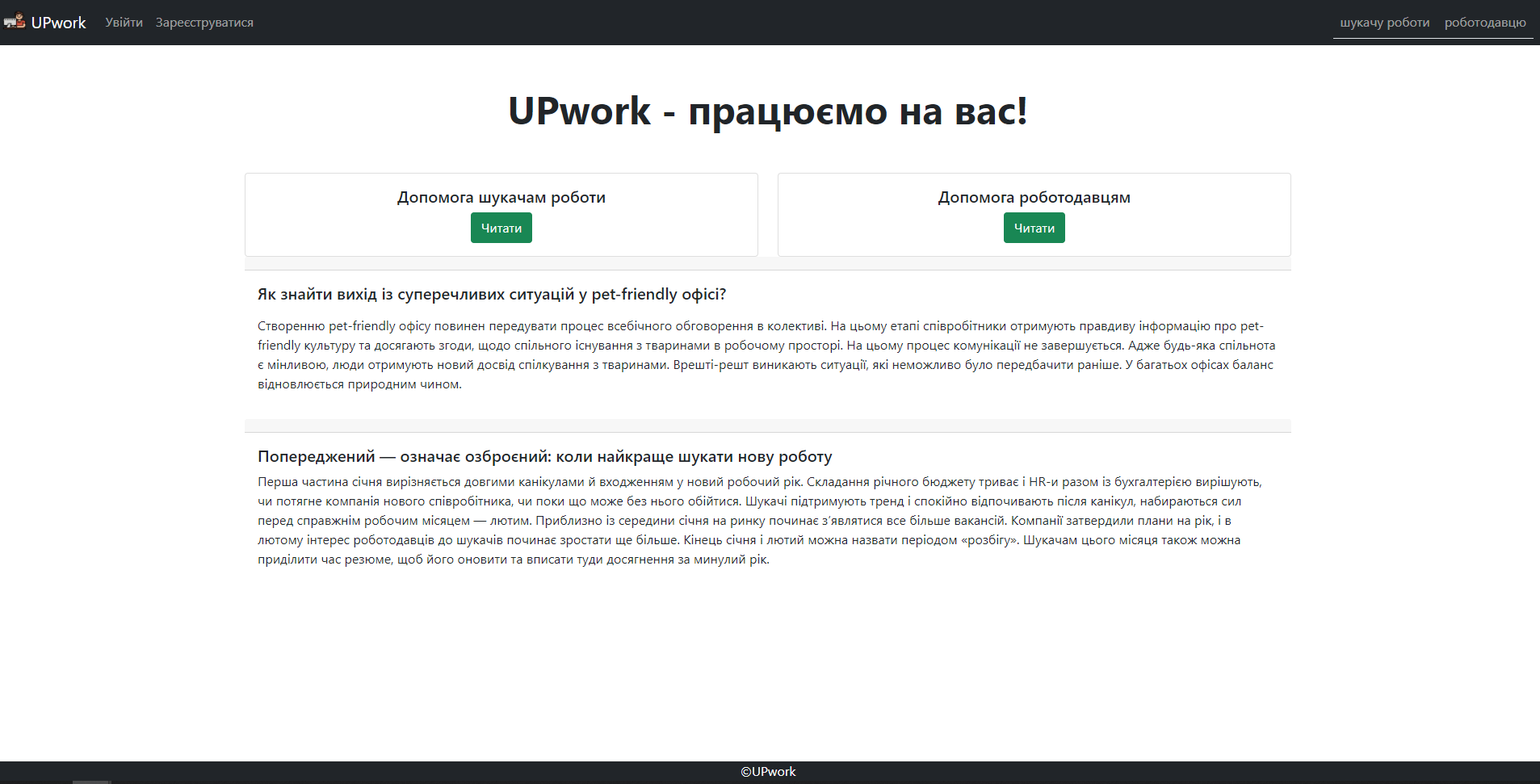
В другому розділі був спроектований алгоритм роботи сайту, визначення основних подій, функцій кожної сторінки. Був розроблений повноцінний сайт, який готовий до тестування. В процесі роботи було побудовано діаграми бази даних та системи класів.

**РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ**

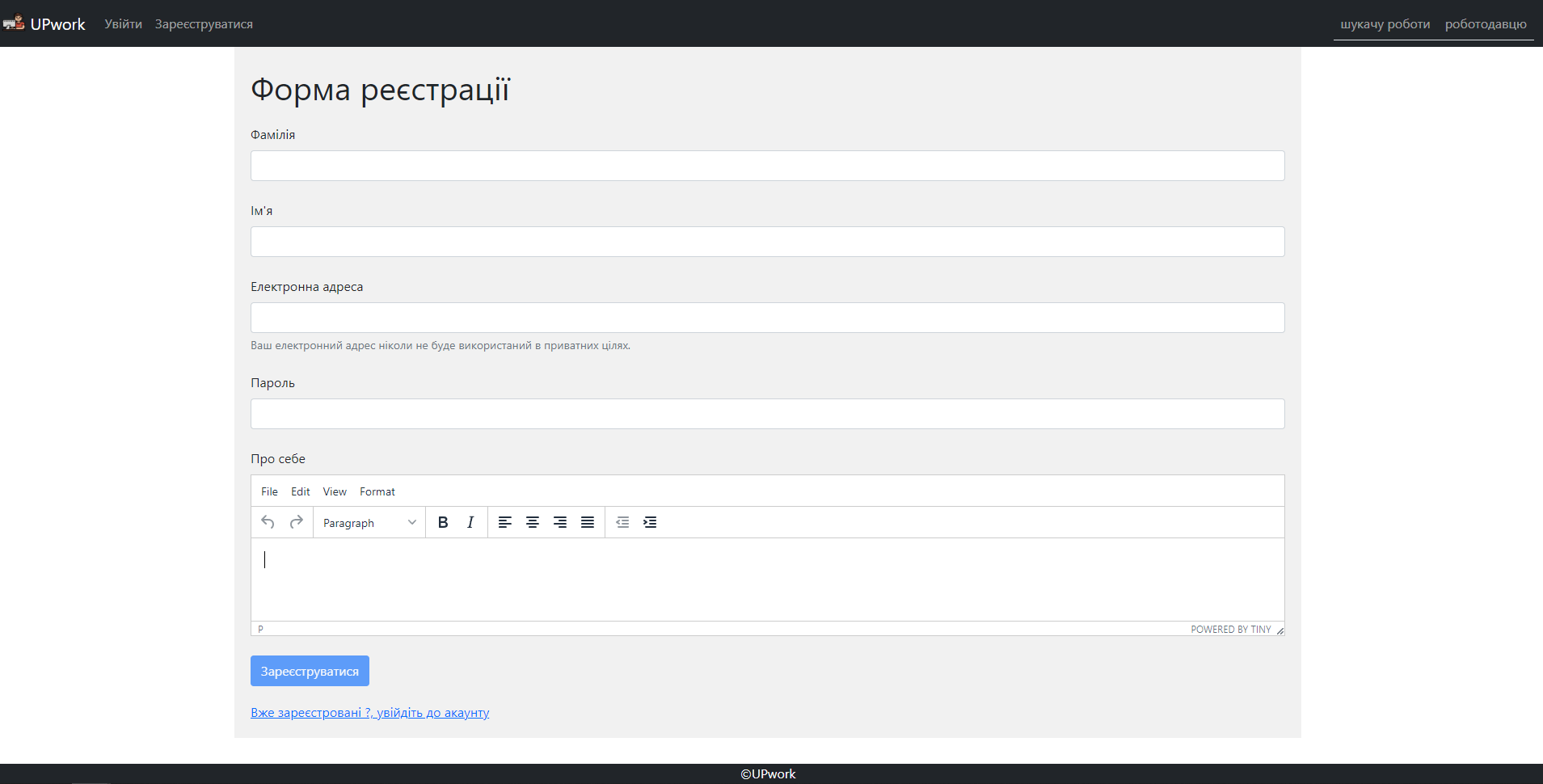
**3.1 Опис роботи з програмним додатком**

Після завершення роботи можна приступити до опису та тестування над сайтом.

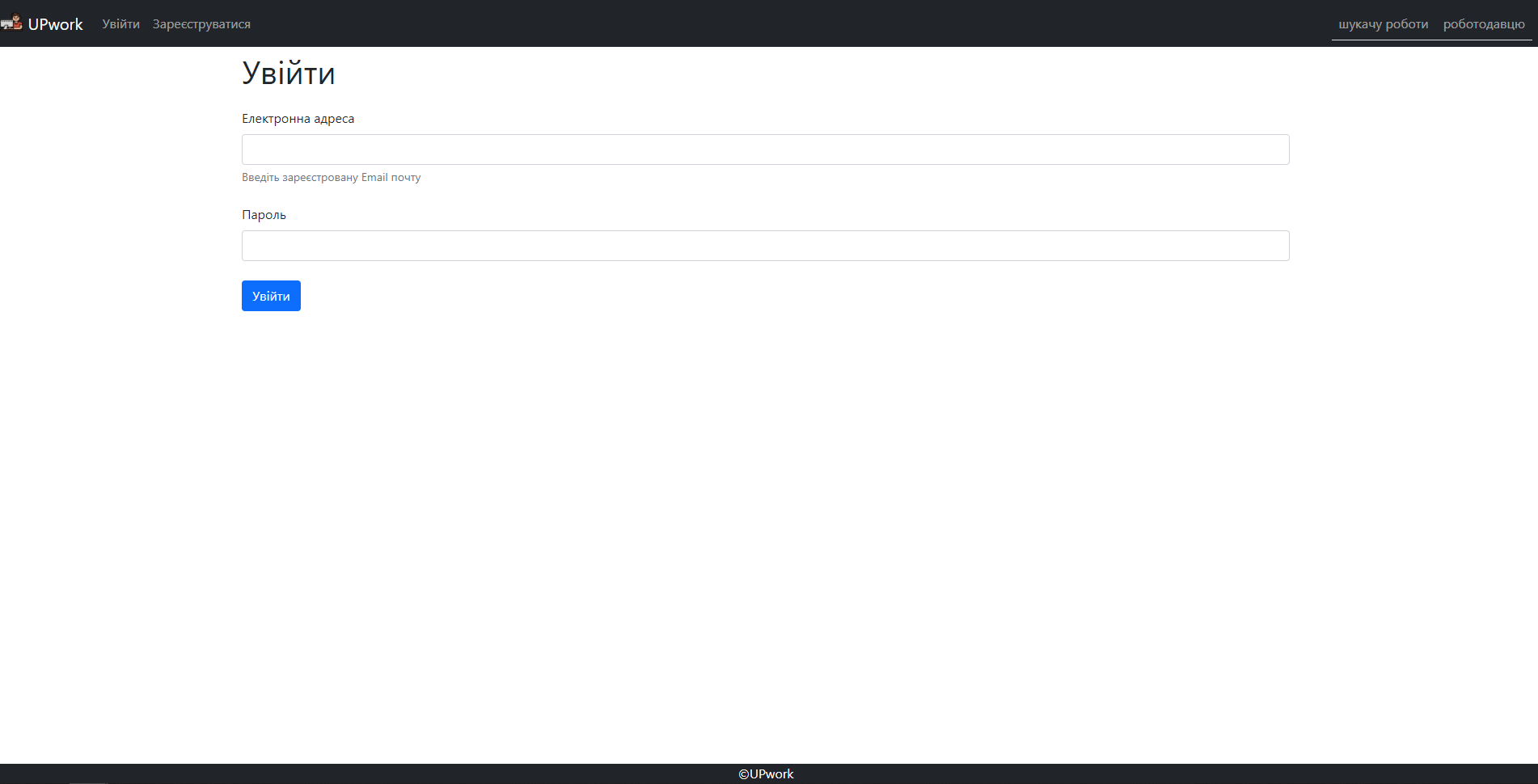
Вперше запустивши сайт, користувач побачить головне меню сайту, на якому присутні новини та інформація для нових користувачів, він матиме змогу ввійти на сайт зареєструвавшись або увійти у вже створений акаунт. Також користувач може переглядати резюме та вакансії. Після аутинтифікації у користувача є можливість створення резюме та вакансій, редагування вже створених.



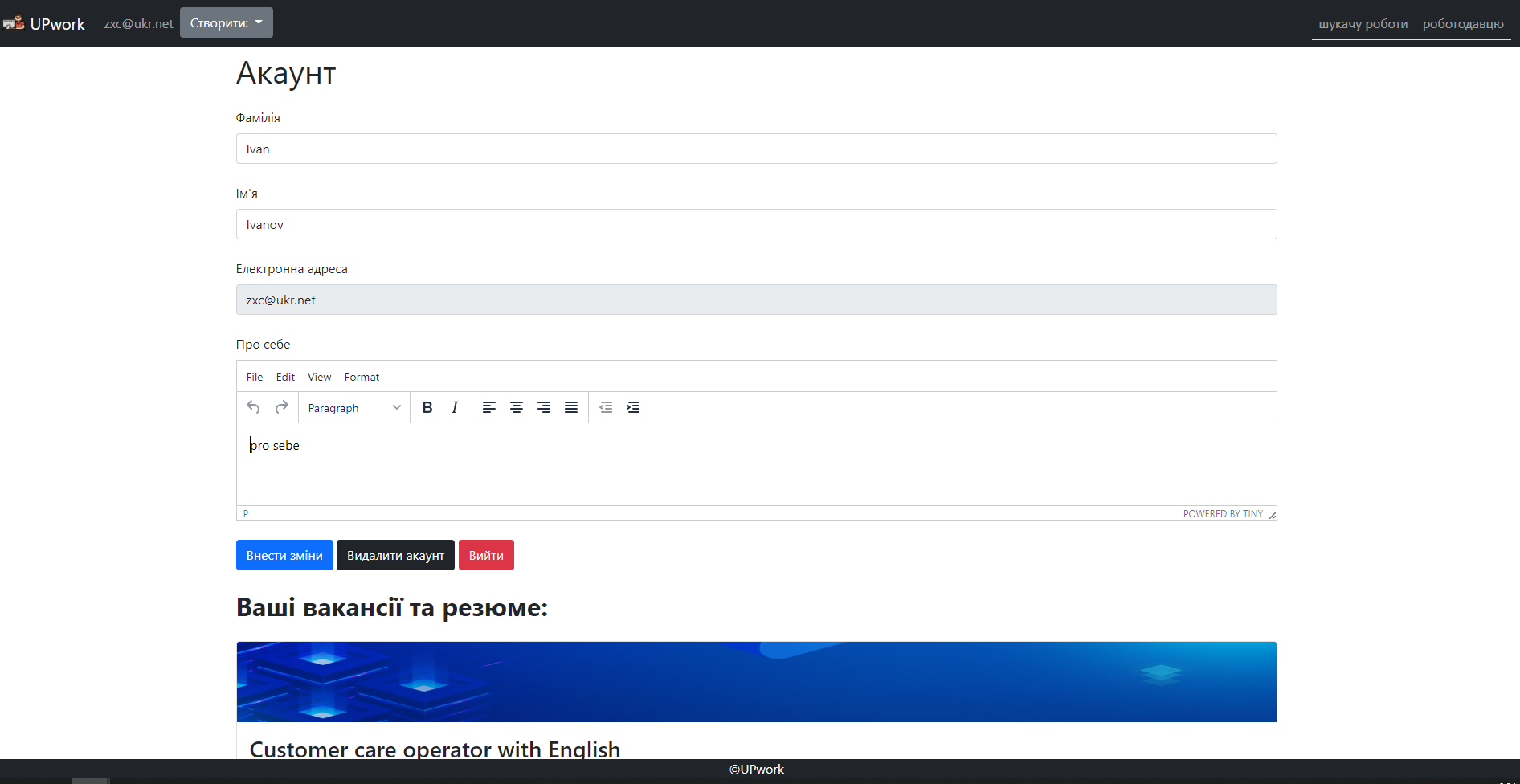
*Рис. 3.1 Головне меню при першому вході*



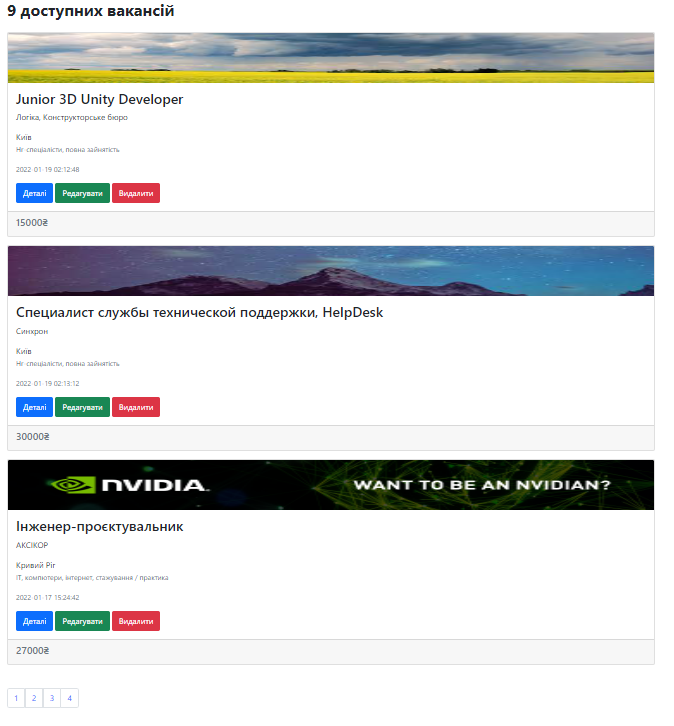
*Рис. 3.2 форма реєстрації на сайті*



*Рис. 3.3 форма входу до кабінету на сайті*



*Рис. 3.4 управління акаунтом*



*Рис. 3.5 приклад контенту на сайті*

**3.2 Тестування роботи програмного забезпечення**

На сайті було реалізовано панель для адміністратора, в якому він може додавати товари, новини, а також їх редагувати і видаляти. Доступ до цих панелей заблокований для інших користувачів. Також під час помилки 404, було вирішено створити окрему сторінку, на яку здійснювалось перенаправлення у випадку введення не коректного URL.

**Висновки до третього розділу**

В третьому розділі було проведено детальний опис роботи з програмним додатком, також створено детальний опис можливих дій користувача. Було проведено повне тестування додатку. Усі помилки, недоліки, які були знайдені в процесі тестування програмного додатку, були прибрані.

**ВИСНОВКИ**

Під час виконання даної курсової роботи я закріпив отримані в результаті навчання протягом семестру знання що до створення сайтів. Також у процесі реалізації програмного продукту поглибив знання мови програмування PHP, JavaScript.

Основною ціллю та задачею моєї курсової роботи було створення сайту порталу для пошуку роботи. Впродовж всього виконання курсового проекту було проведено аналіз задачі, обрано способи та методи її вирішування. Зроблено детальний аналіз уже існуючих конкурентів на ринку, за допомогою якого було зрозуміло який функціонал необхідний у власному сайті. Спроектовано послідовність роботи додатку та спроектовано алгоритми для її роботи.

Створено графічний інтерфейс додатку та написано програмний код.

Детальний опис роботи сайту можна використовувати як «мануал» для користувачів, це допоможе швидше адаптуватись до швидкої роботи з цим сайтом. Впродовж тестування сайту декілька раз доводилось виправляти помилки в роботі додатку та дрібні «помилки», для подальшого комфорту у взаємодії з програмним додатком.

Робота над курсовим проектом відбувалась близько місяця. Результатом роботи є повністю функціонуючий сайт з базою даних, який розкриває тему проекту курсової роботи вказану раніше, та має всі потрібні для використання функції. В кінці роботи написано пояснювальну записку проекту обсягом 25 сторінок, без врахування додатків.

Програмний продукт повністю відповідає вимогам, заданим на початку курсового проекту.

**СПИСОК ЛІТЕРАТУРИ**

1. Классы и объекты , 2013. Режим доступу:

https://www.php.net/manual/ru/language.oop5.php

2. admin. MS Access и C# [Електронний ресурс] / admin. – 2018. – Режим

доступу до ресурсу: https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-sbd.html

3. admin. Все операции с БД в графическом приложении [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/adonet/3.5.php

4. Паттерны для новичков: MVC vs MVP vs MVVM [електронний ресурс] - 2006. Режим доступу: <https://habr.com/ru/post/215605/>

5. Тим Кедлек. “Адаптивный дизайн. Делаем сайты для любых устройств”

[електронний ресурс] - 2013. Режим доступу:

https://www.ozon.ru/context/detail/id/21098804/

6. Документація бібліотеки JQuery [Електронний ресурс] // JQuery incorp. –

2016. – Режим доступ: https://jquery-docs.ru

7. Выразительный JavaScript [електронний ресурс] - 2017. Режим доступу:

https://legacy.gitbook.com/book/karmazzin/eloquentjavascript\_ru/details

8. Основы наследования [електронний ресурс] - 2015. Режим доступу:

https://professorweb.ru/my/ level7/7\_1.php

9. Плюси і мінуси обєктно-орієнтованого програмування [електронний

ресурс] - 2015. Режим доступу: http://damp.biz/plyusi-i-minusi-obyektnooriyentovanogo-programuvannya/

10. SQL [електронний ресурс] - 2016. Режим доступу:

https://uk.wikibooks.org/wiki/SQL

11. Создание SQL запросов [електронний ресурс] - 2016. Режим доступу:

https://www.politerm.com/zuludoc/spatial\_query\_sql.htm#\_sql\_overview

**ДОДАТКИ**

Лістинг Controller класів:

<?php  
  
class Controller\_404 extends Controller  
{  
 function action\_index()  
 {  
 $this->view->generate('404\_view.php', 'template\_view.php');  
 }  
}

?php  
  
class Controller\_Admin extends Controller  
{  
  
 function action\_index()  
 {  
 session\_start();  
  
 /\*  
 Для простоты, в нашем случае, проверяется равенство сессионной переменной admin прописанному  
 в коде значению — паролю. Такое решение не правильно с точки зрения безопасности.  
 Пароль должен храниться в базе данных в захешированном виде, но пока оставим как есть.  
 \*/  
 if ( $\_SESSION['admin'] == "12345" )  
 {  
 $this->view->generate('admin\_view.php', 'template\_view.php');  
 }  
 else  
 {  
 session\_destroy();  
 Route::ErrorPage404();  
 }  
  
 }  
  
 // Действие для разлогинивания администратора  
 function action\_logout()  
 {  
 session\_start();  
 session\_destroy();  
 header('Location:/');  
 }  
  
}

<?php  
  
class Controller\_Main extends Controller  
{  
 protected $user;  
 public function \_\_construct()  
 {  
 $this->model = new model\_main();  
 $this->view = new View();  
 $this->user = $this->model->GetCurrentUser();  
 }  
  
 function action\_index()  
 {  
 $data = $this->model->getcards();  
 $this->view->generate('main/main\_view.php', 'template\_view.php', $data);  
 }  
  
 function action\_add(){  
 $us = $this->user;  
 if($us != 11){  
 $this->view->generate('forbidden\_view.php', 'template\_view.php');  
 }  
 else{  
 $this->model->addcard();  
 $this->view->generate('main/add\_view.php', 'template\_view.php');  
 }  
 }  
 function action\_edit(){  
 $id = $\_GET['id'];  
 $data = $this->model->getcardID($id);  
 $us = $this->user;  
 if($us == 11){  
 $this->model->editcard($id);  
 $this->view->generate('main/edit\_view.php', 'template\_view.php', $data);  
 }  
 else {  
 $this->view->generate('forbidden\_view.php', 'template\_view.php');  
 }  
 }  
 function action\_delete(){  
 $id = $\_GET['id'];  
 $data = $this->model->getcardID($id);  
 $us = $this->user;  
 if($us == 11){  
 $this->model->deletecard($id);  
 $this->view->generate('main/main\_view.php', 'template\_view.php');  
 }  
 else {  
 $this->view->generate('forbidden\_view.php', 'template\_view.php');  
 }  
 }  
  
 function action\_employment(){  
 $this->view->generate('main/employment\_view.php', 'template\_view.php');  
 }  
 function action\_searchjob(){  
 $this->view->generate('main/searchjob\_view.php', 'template\_view.php');  
 }  
  
}

<?php  
  
class Controller\_user extends Controller  
{  
 function \_\_construct()  
 {  
 $this->model = new model\_user();  
 $this->view = new View();  
 }  
 function action\_register()  
 {  
 $this->model->get\_reg();  
 $this->view->generate('user/register\_view.php', 'template\_view.php');  
 }  
 function action\_login()  
 {  
 $this->model->get\_login();  
 $this->view->generate('user/login\_view.php', 'template\_view.php');  
 }  
 function action\_logout()  
 {  
 $this->model->log\_out();  
 $this->view->generate('main\_view.php', 'template\_view.php');  
 }  
 function action\_account()  
 {  
 session\_start();  
 if(isset($\_SESSION['id'])){  
 $data1 = $this->model->getvacancybyUSERID($\_SESSION['id']);  
 $data2 = $this->model->getresumebyUSERID($\_SESSION['id']);  
 $this->model->get\_changes();  
 $this->view->generate('user/account\_view.php', 'template\_view.php', $data1, $data2);  
 }  
 else{  
 $this->view->generate('forbidden\_view.php', 'template\_view.php');  
 }  
 }  
 function action\_deleteAc()  
 {  
 $this->model->delete\_acc();  
 $this->view->generate('user/login\_view.php', 'template\_view.php');  
 }  
  
}

<?php  
  
class Controller\_user extends Controller  
{  
 function \_\_construct()  
 {  
 $this->model = new model\_user();  
 $this->view = new View();  
 }  
 function action\_register()  
 {  
 $this->model->get\_reg();  
 $this->view->generate('user/register\_view.php', 'template\_view.php');  
 }  
 function action\_login()  
 {  
 $this->model->get\_login();  
 $this->view->generate('user/login\_view.php', 'template\_view.php');  
 }  
 function action\_logout()  
 {  
 $this->model->log\_out();  
 $this->view->generate('main\_view.php', 'template\_view.php');  
 }  
 function action\_account()  
 {  
 session\_start();  
 if(isset($\_SESSION['id'])){  
 $data1 = $this->model->getvacancybyUSERID($\_SESSION['id']);  
 $data2 = $this->model->getresumebyUSERID($\_SESSION['id']);  
 $this->model->get\_changes();  
 $this->view->generate('user/account\_view.php', 'template\_view.php', $data1, $data2);  
 }  
 else{  
 $this->view->generate('forbidden\_view.php', 'template\_view.php');  
 }  
 }  
 function action\_deleteAc()  
 {  
 $this->model->delete\_acc();  
 $this->view->generate('user/login\_view.php', 'template\_view.php');  
 }  
  
}