МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ  
Факультет інформаційних технологій і систем  
Кафедра програмного забезпечення автоматизованих систем

Курсова робота

з дисципліни «Професійна практика програмної інженерії»  
тема: «Web-додаток система тестування рівня кваліфікації працівника.

Front-end Web-додатку»  
Напрям підготовки 6.050103 програмна інженерія  
Пояснювальна записка

Перевірив:

Ст. викладач

Плакасова Ж. М.

Ст. викладач

Півень О. Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оцінка

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

підписи

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

підписи

Виконав:

Студент групи ПЗ-154

Гаврилюк В.Є.

Залікова книжка №1539

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Черкаси, 2019

# **ЗМІСТ**

Зм.

Лист

№ документа

Підпис

Дата

Розроб.

Керівник

Н.контр.

Затв.

Гаврилюк В.Є.

Плакасова Ж. М.

Півень О. Б.

ЧДТУ 191539.003 ПЗ

«Web-додаток система тестування рівня кваліфікації працівника.

Front-end Web-додатку»

Пояснювальна записка

Літ.

Лист

Листів

НД

**ФІТІС,**

кафедра ПЗАС, ПЗ-154

3

33

[**ЗМІСТ 3**](#_Toc7773557)

[**СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ 4**](#_Toc7773558)

[**ВСТУП 5**](#_Toc7773559)

[**1 ПОСТАНОВКА ЗАДАЧІ 7**](#_Toc7773560)

[1.1 Індивідуальне завдання 7](#_Toc7773561)

[1.2 Технічне завдання 7](#_Toc7773562)

[**2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8**](#_Toc7773563)

[**3 ДОСЛІДЖЕННЯ АНАЛОГІВ 10**](#_Toc7773564)

[**4 РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ 12**](#_Toc7773565)

[**5 РОЗРОБКА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ 17**](#_Toc7773566)

[**6 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ 22**](#_Toc7773567)

[6.1 Операційна система 22](#_Toc7773568)

[6.2 Мова програмування і середовище розробки 22](#_Toc7773569)

[6.3 Необхідні вимоги до конфігурації ЕОМ 23](#_Toc7773570)

[**7 ТЕОРЕТИЧНІ ВІДОМОСТІ 24**](#_Toc7773571)

[7.1 Hypertext Markup Language 24](#_Toc7773572)

[7.2 Cascading Style Sheets 26](#_Toc7773573)

[7.3 Angular 29](#_Toc7773574)

[7.4 TypeScript 30](#_Toc7773575)

[**ВИСНОВКИ 32**](#_Toc7773576)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 33**](#_Toc7773577)

# **СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ**

|  |  |  |
| --- | --- | --- |
| CSS | - | Cascading Style Sheets |
| SCSS | - | Sassy Cascading Style Sheets |
| SQL | - | Structured query language |
| HTML | - | Hypertext Markup Language |
| URL | - | Uniform Resource Locator |
| XML | - | Extensible Markup Language |

# **ВСТУП**

Сьогодні в галузі ІТ існує дуже багато компаній із числом працівників понад тисячу і значно вище. Такі компанії дуже активно приймають на роботу нові кадри, але через високе завантаження відділу найму вони витрачають багато ресурсів на прості завдання, а саме попереднє тестування кандидата. Дана система має на меті спростити процес прийому на роботу нового працівника та зменшити витрати пов’язані з цим, за рахунок проведення попереднього стандартизованого тестування кандидата.

Стандартизований тест або іспит являє собою тестування, яке проводиться та перевіряється заздалегідь установленим, або «стандартним», шляхом. Стандартизовані тести розроблені таким чином, що питання, умови проведення, оцінювання і тлумачення заздалегідь погоджені, процеси проведення і оцінювання визначені наперед стандартним чином. Стандартне тестування називають найбільш значним соціальним внеском сучасної психології.

Будь-який тест, в якому той же самий тест дається таким же чином для всіх тестованих, являє собою стандартизований тест. Стандартизовані тести — це не обов'язково тести з високими ставками, тести з обмеженим часом, або тести з декількома варіантами відповідей. Протилежністю стандартизованого тестування є нестандартизоване тестування, в якому або істотно різні випробування наведені до різних тестованих, або той же самий тест призначається при істотно різних умовах (наприклад, одній групі дозволено набагато менше часу, щоб завершити випробування, ніж наступній групі) або оцінюватися по-різному (наприклад, та же відповідь вважається правильним для одного студента, але неправильно для іншого студента). Підрахунок очок есе робиться швидко і не-вихователів. Забивши нариси люди, які не приймають їх час і отримують бонуси є пасткою у виграші.

Стандартизовані тести сприймаються як більш справедливі, ніж не-стандартизовані тести, тому що кожен отримує той же самий тест і ту ж систему класифікації. Це є більш справедливим і більш об'єктивним, ніж система, в якій деякі студенти отримують більш простий тест і інші отримують більш важке випробування. Узгодженість також дозволяє більш надійне порівняння результатів у всіх тестованих, тому що кожен приймає один і той же тест.

Актуальність даної теми полягає у тому, що компаніям потрібно перевіряти рівень працівників, які в них працюють не витрачаючи багато часу на різні багато часові опитування. Також, даний web-додаток полегшить та спростить процедуру співбесіди та визначення рівня працівника.

Метою є розробка web-додатку, яким будуть користуватись компанії. Вони зможуть з легкістю добавити в додаток свої бібліотеки питань та відповідей до них, та використовувати їх для опитування різного рівня працівників. Як результат кожен додаток буде мати свою базу даних, та свої унікальні питання, тому що як ми знаємо, рівень оцінювання працівників у кожного свій.

У всьому світі з ринковою економікою існував ринок праці програмістів, який відрізнявся високою конкуренцією. Отже, він завжди був здатний регулювати себе сам, не кажучи вже про те, що процес ліцензування — досить коштовна процедура. Хто має платити за неї? Що робити з тими, хто її не пройшов? Є й багато інших проблем, пов’язаних із проблемою ліцензування.

Об’єктом дослідження курсової роботи є інтерфейс системи попереднього тестування працівників перед основною співбесідою «Test system». Забезпечення користувача всім необхідним інструментарієм у форматі web-додатку.

Предметом дослідження даної курсової роботи є взаємодія користувача із інтерфейсом веб-додатку, можливість створення нових тестів, перегляд резултатів вже пройдених та, власне, самого тестування.

1. **ПОСТАНОВКА ЗАДАЧІ**

## **1.1 Індивідуальне завдання**

Завданням даної роботи є реалізація web-додатку на тему «Web-додаток система тестування рівня кваліфікації працівника. Front-end Web-додатку».

Програма повинна бути простою, інтуїтивно зрозумілою та комфортною у використанні. Програма повинна відповідати системним вимогам.

Додаток має містити базу даних для зберігання користувачів, питань, відповідей на питання.

Також, створенням тестів для користувачів повинна займатись людина з правами admin.

Прості користувачі можуть тільки проходити тести та дивитись свій результат.

* 1. **Технічне завдання**

Розробка програмного продукту має містити наступні пункти.

1 Ім'я додатку (назва домену).

2 Назва додатку.

3 Призначення додатку(мета створення додатку).

4 Мова додатку.

5 Основні ключові слова.

6 Передбачувана вікова аудиторія додатку.

7 Кількість сторінок додатку.

8 Зрозумілий інтерфейс.

9 Оптимізований додаток.

10 Діапазон дозволу моніторів, на яких буде проглядатися додаток.

1. Підтримка браузерами.
2. **АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

**Аналіз області застосування програмного продукту.**

Сьогодні в галузі ІТ існує дуже багато компаній із числом працівників понад тисячу і значно вище. Такі компанії дуже активно приймають на роботу нові кадри, але через високе завантаження відділу найму вони витрачають багато ресурсів на прості завдання, а саме попереднє тестування кандидата. Дана система має на меті спростити процес прийому на роботу нового працівника та зменшити витрати пов’язані з цим, за рахунок проведення попереднього стандартизованого тестування кандидата.

Стандартизований тест або іспит являє собою тестування, яке проводиться та перевіряється заздалегідь установленим, або «стандартним», шляхом. Стандартизовані тести розроблені таким чином, що питання, умови проведення, оцінювання і тлумачення заздалегідь погоджені, процеси проведення і оцінювання визначені наперед стандартним чином. Стандартне тестування називають найбільш значним соціальним внеском сучасної психології.

**Аналіз потреб користувача**

Світова індустрія інформаційних технологій перебуває на швидкості, щоб досягти $ 5 трлн. У 2019 році, повідомляє дослідницька консультація IDC. Величезна індустрія є функцією багатьох тенденцій, що обговорюються в цьому звіті. Економіка, робота і особисте життя стають все більш цифровими, більш пов'язаними і все більш і більш автоматизованими. Хвилі інновацій будуються з плином часу, приводячи в дію двигун росту технологій, який, здається, знаходиться на порозі ще одного великого стрибка вперед.

За останні два роки кількість спеціалістів в топ-50 найбільших ІТ-компаніях виросла більше ніж на третину — з 43 тис. до 58 тис.

З липня по січень кількість фахівців зросла на 4 731 (11.6 %) в топ-25 та 5 797 (11%) в топ-50 порівняно з першим півріччям 2018-го. Темпи зростання топ-25 стали рекордними за останні п’ять років.(рисунок 1; рисунок 2;)

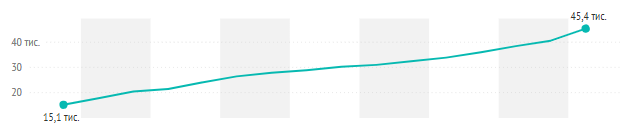


Рисунок 2.1 – Зростання загальної кількості спеціалістів в 25 найбільших ІТ-компаніях України

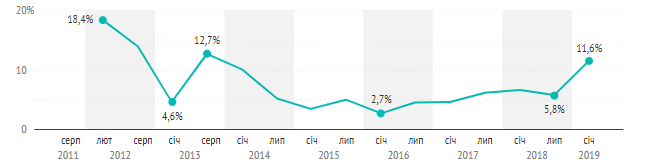


Рисунок 2.2 – Відносні показники темпів зростання

Ми з впевненістю можемо сказати, що компанії будуть набирати більше і більше працівників. Як наслідок, вони будуть в пошуку автоматизації та спрощенні співбесід та перевірки або підвищення кваліфікації персоналу. (рисунок 3)

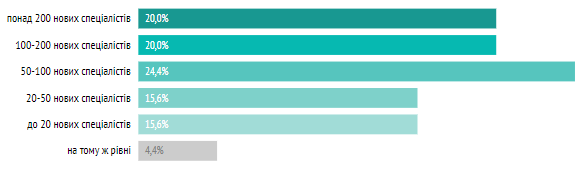


Рисунок 2.3 – Плани компаній щодо розширення персоналу на найближчі півроку

1. **ДОСЛІДЖЕННЯ АНАЛОГІВ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Назва** | **Реєстрація** | **Можливість отримання результату** | **Групування по темам** | **Автоматизованість, проходження за певний час** | **Можливість вибрати рівень** |
| jsehelper | - | - | + | - | - |
| quizful | + | + | + | + | - |
| proghub | + | + | - | + | - |
| intester | + | + | - | + | - |

Таблиця 3.1 – Дослідження аналогів

**Jsehelper**

Jsehelper – це блог, на якому можна знайти набір питань, для технології і відповіді на них. Але не має реєстрації, та можливості вибирати рівень особи та генерацію питань. Також, великий мінус є те, що не має можливості вести статистику відповідей.

**Quizful**

Quizful – це повноцінний онлайн сервіс для проходження різних тестів. Можна вибрати тести, тему, та дивитись статистику відповідей. Мінусом даного сервісу є неможливість генерації рандомної кількості питань, ви можете створити тести, але в них буде фіксована однакова кількість питань.

**Proghub**

Proghub – онлайн сервіс який тільки з’явився. Можна зареєструватись, і проходити різні тести по наявним технологіям. Проблема полягає в тому, що на даному сервісі, не можна вибрати рівень, або кількість питань. Ви просто вибираєте технологію, і проходите відповідний тест.

**Intester**

Intester –онлайн сервіс для перевірки знань. Є можливість реєстрації, та проходження тестів по певних категоріях. Ви не можете вибрати рівень кваліфікації, та кількість тем, які будуть в даних тестах.

Переваги даного проекту, після аналізу даного матеріалу.

1. Реєстрація.
2. Можливість подивитись результат.
3. Групування питань по тематиці.
4. Вибір рівня кваліфікації для якої буде генеруватись список питань.
5. Створенням тестів для працівників будуть займатись відповідні люди, і прості працівники не можуть ніяк впливати на результат, або генерувати тести для себе.

Недоліки.

1. Потрібна реклама.
2. Великий проект не можливо написати без команди.
3. **РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ**

Даний розділ описує усю інформацію, яку потрібно враховувати при розробці програмного продукту. Даний розділ фіксує мету та вимоги, до програмного продукту. На основі інформації, що приводиться в даному документі, відбувається реалізація програмного продукту.

Мета створення інтерфейсу web-додатку. Створити зручний та простий інтерфейс web-додатку для стандартизованого тестування кандидата на роботу. Додати можливість створення нового тесту та перегляду вже пройдених тестів. Додати блок із основною статистичною інформацією. Додати окрему сторінку для проходження тесту. Сторінки повинні бути оптимізованими та містити як найменше зайвої інформації. Кольорова гама має бути підібрана вдало, щоб кольори легко сприймались.

Призначення. Цей web-додаток призначений для спрощення прийому на роботу нових працівників. Перед основною співбесідою кандидат буде проходити тестування, яке дозволить відсіяти значну частину некомпетентних та ще не готових робітників.

Цей web-додаток допоможе компаніям зменшити навантаження на HR відділи та зменшити супутні витрати, пов’язані з взяттям на роботу нових працівників.

Цільова аудиторія web-додатку. Цільовою аудиторією є компанії, що мають велике навантаження на свої HR відділи та прагнуть зменшити його і супутні витрати. Цим додатком будуть користуватися інтерв’ювери, які в основному є молодими людьми у віці від 25 до 40 років, тому додаток повинен бути простим і без зайвого нагромадження.

Мови web-додатку. Web-додаток створено на англійській мові. Тому його можуть використовували люди з будь-якої країни, які володіють базовими знаннями англійської мови.

Структура web-додатку

Таблиця 4.1

Структура web-сайту

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Роут сторінки** | **Зміст сторінки, розділу** | **Примітка** |
| 1 | /  Login page | Сторінка має лише центральний блок. Він має два поля та кнопку. Користувач має ввести свої логін та пароль, щоб увійти в цю систему. | Сторінка має вбудовану валідацію даних, що вводяться користувачем. |
| 2 | /home  Home page | Сторінка поділена на три блоки. Верхній блок має логотип та кнопку для показу або приховування лівого навігаційного меню. Лівий блок складається із навігаційного меню. На правому блоці розміщена основна статистична інформація додатку. | Сторінка містить ліве навігаційне меню. |
| 3 | /home/surveys-list  Completed surveys list page | Сторінка поділена на три блоки. Верхній блок має логотип та кнопку для показу або приховування лівого навігаційного меню. Лівий блок складається із навігаційного меню. На правому блоці розміщена таблиця із списком вже пройдених тестувань. | Сторінка містить ліве навігаційне меню. |

Продовження табл. 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | /home/survey-info?id=[*id*]  View information about completed survey page | Сторінка поділена на три блоки. Верхній блок має логотип та кнопку для показу або приховування лівого навігаційного меню. Лівий блок складається із навігаційного меню. На правому блоці розміщена детальна інформація по конкретному тесту – є всі відповіді і позначки про їх вірність. | Сторінка містить ліве навігаційне меню. |
| 5 | /home/create-survey  Create new survey page | Сторінка поділена на три блоки. Верхній блок має логотип та кнопку для показу або приховування лівого навігаційного меню. Лівий блок складається із навігаційного меню. На правому блоці розміщена форма для створення нового тестування. Під нею розміщена таблиця із посиланнями на щойно створені тестування. | Сторінка містить ліве навігаційне меню. Також тут реалізований пошук осіб для тестування з допомогою методики Typehead. |
| 6 | /survey?token=[ *token*]&id=[*id*] | Сторінка поділена на два блоки. Верхній блок має логотип додатку. | Сторінка має захист від повторного проходження вже складеного тесту. |

Продовження табл. 4.1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | На іншому розміщені блоки із питаннями та варіантами відповідей, які потрібно вибрати, або полями, куди потрібно записати відповідь. |  |
| 7 | /[*будь-яке відсутнє в системі посилання*]  404 page | Сторінка має лише блок із інформацією про те, що посилання на сторінку є невірним та нерозпізнаним системою. | Під інформаційним блоком є посилання для повернення на попередню сторінку. |

**Вимоги до функціональних характеристик**

В даному пункті, наведено всі вимоги до функціональних характеристик.

1. Створення нового тесту.
2. Можливість перегляду основної статистичної інформації системи.
3. Можливість перегляду списку всіх тестувань.
4. Можливість перегляду детальної інформації по окремим тестуванням.
5. Можливість проходження тестування в окремому вікні.

**Підтримка браузерів**

Web-додаток має підтримку всіх браузерів.

Наприклад:

1. Google Chrome;
2. Firefox;
3. Safari;
4. Opera.

**Вимоги до безпеки**

Безпека це одна з найважливіших аспектів веб-додатків. Якщо безпека вразлива, то весь додаток не є надійним.

Я виділив декілька пунктів, які було зроблено для того, щоб безпека була на доступному рівні.

Для того, щоб кожен мав певну безпеку, було зроблено декілька вимог.

1. Поля пошуку фільтруються від SQL команд.
2. Поля введення інформації фільтруються від SQL команд.
3. Поля пошуку фільтруються від спеціальних символів.
4. Зашифрована передача даних між користувачем та сервером.

# **РОЗРОБКА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ**

Користувацький інтерфейс забезпечує взаємодію між користувачем та додатком. Це важлива складова веб-додатку, оскільки являється єдиним способом для користувачів взаємодіяти із додатком.

Всі користувачі мають різний досвід та навички використання web-додатків. Саме тому інтерфейс повинен забезпечувати зручність користування системою для користувача із будь-яким рівнем комп’ютерної грамотності. Але разом із тим забезпечувати приємний вигляд та відповідати сучасним нормам та стандартам.

Дизайн веб-сайту повинен відповідати усім сучасним нормам та напрямкам у веб-дизайні, бути простим й одночасно сучасним та інформативним. Дизайн необхідно виконати в світлій кольоровій гамі. Дизайн повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу).

Інтерфейс користувача повинен забезпечувати наглядне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації та швидкий і логічний перехід між розділами.

Навігаційні елементи повинні забезпечувати однозначне розуміння користувачами їх змісту: лінки на сторінки повинні містити заголовки, умовні позначення - відповідати загальноприйнятим. Графічні елементи навігації повинні містити альтернативний підпис.

Під час розробки було створено макет інтерфейсу користувача для головних компонентів. Кольорова палітра та стиль оформлення готового програмного продукту мають відповідати значенням в макеті (Рисунок 5.1; Рисунок 5.2; Рисунок 5.3; Рисунок 5.4; Рисунок 5.5; Рисунок 5.6; Рисунок 5.7) Розміщення елементів може відрізнятися.

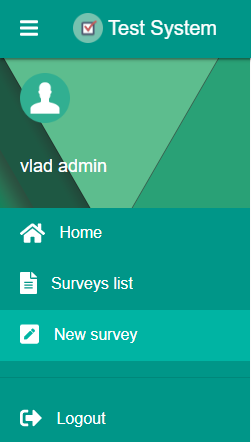


Рисунок 5.1 – макет лівої навігаційної панелі для всіх сторінок

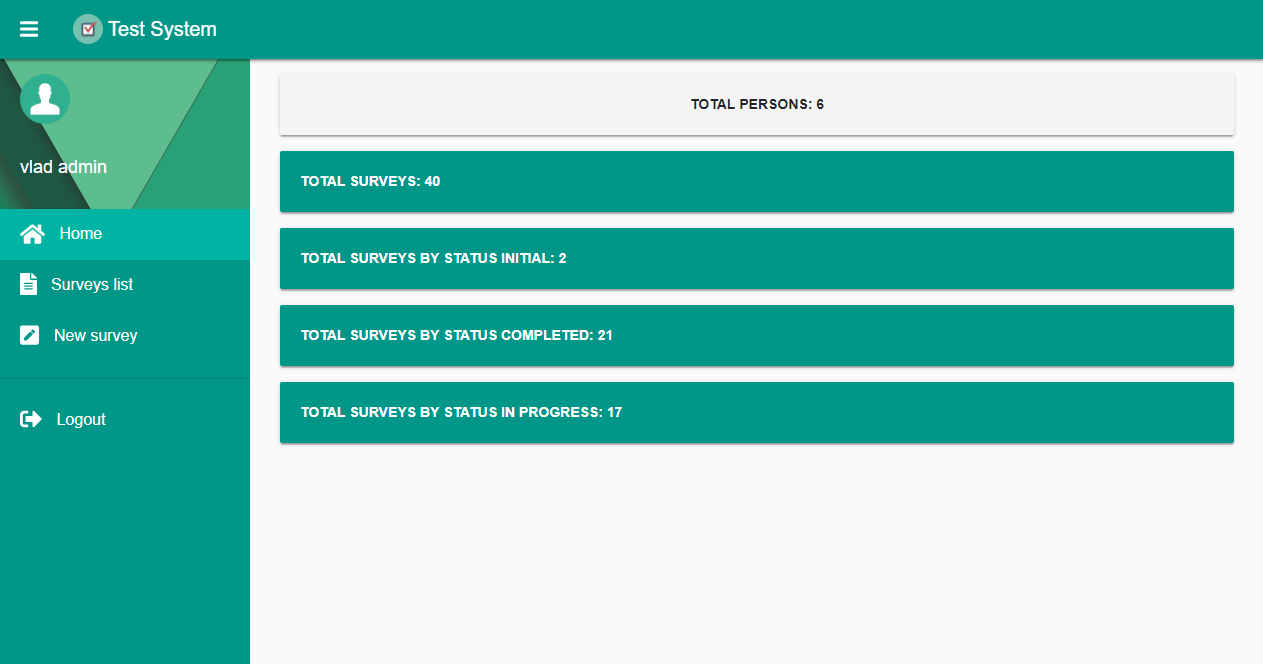


Рисунок 5.2 – Макет головної сторінки із основною статистичною інформацією системи

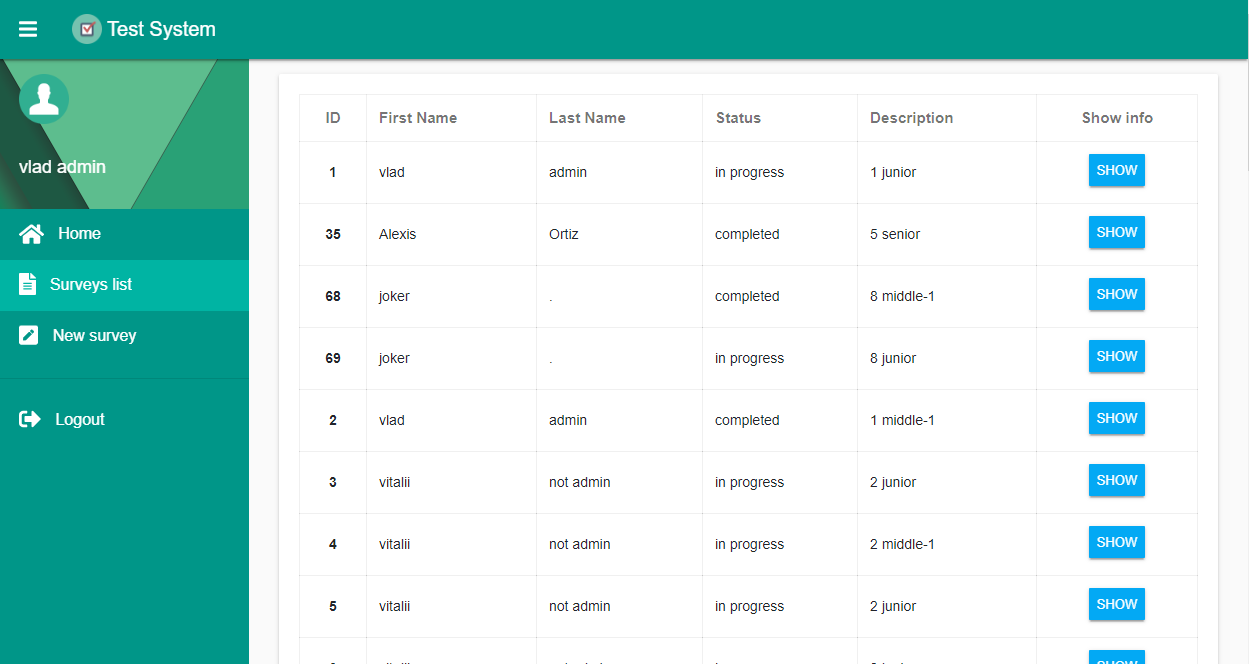


Рисунок 5.3 – Макет сторінки із списком всіх створених тестувань

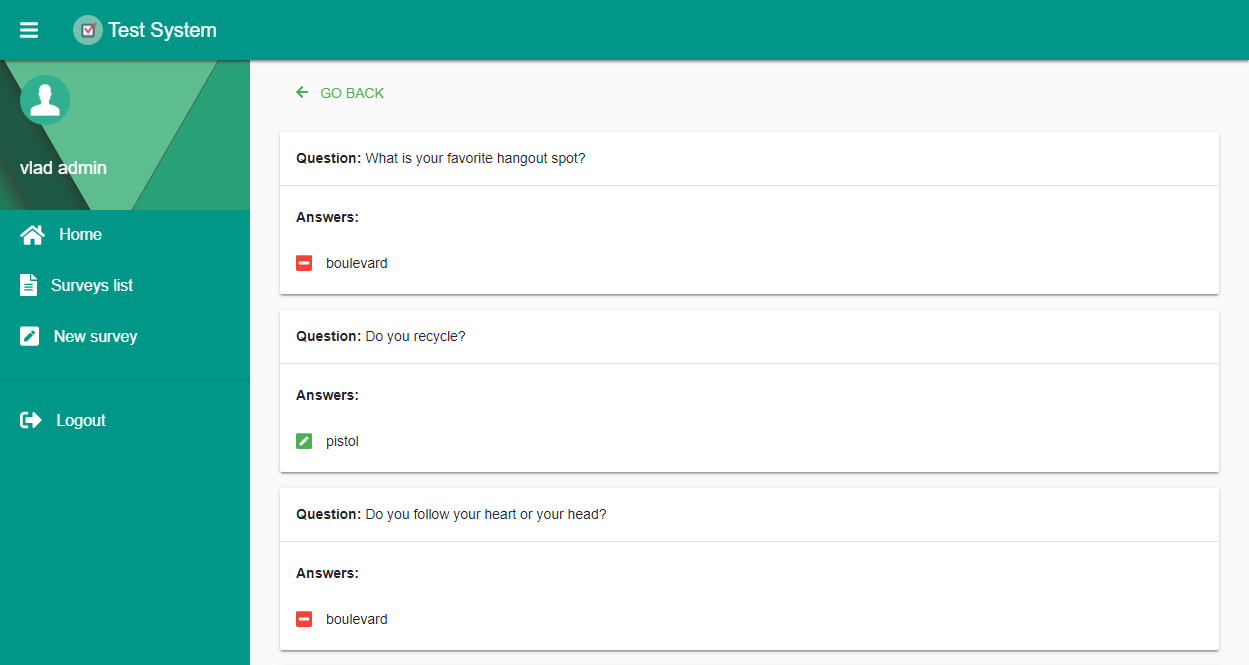


Рисунок 5.4 – Макет сторінки із детальною інформацією по окремому тесту

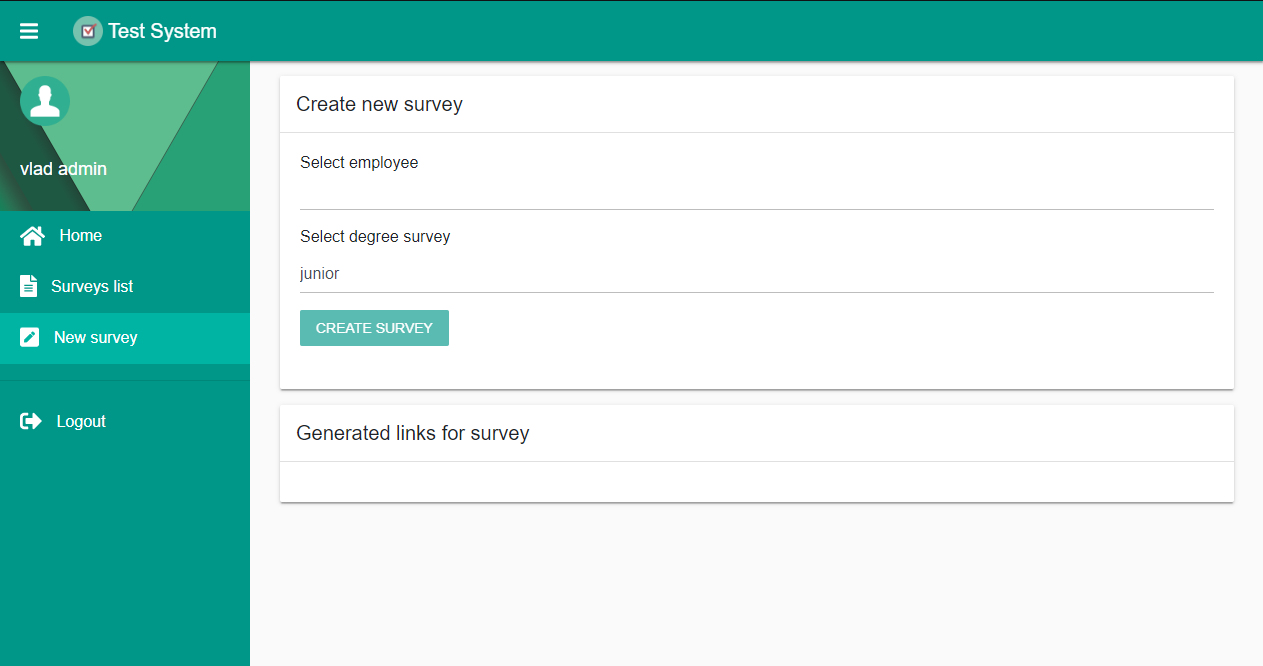


Рисунок 5.5 – Макет сторінки по створенню нового тестування

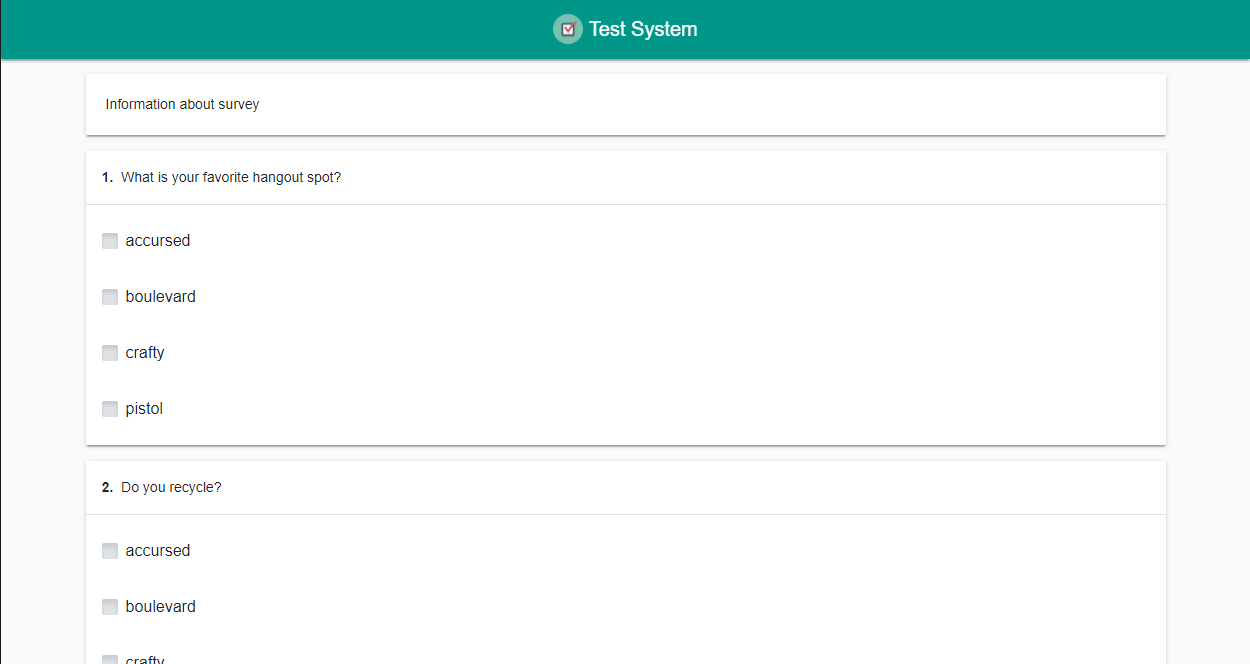


Рисунок 5.6 – Макет сторінки проходження тесту

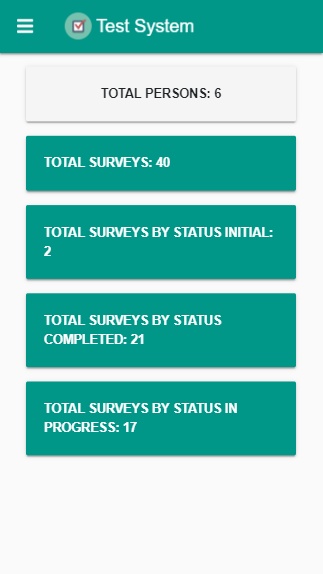


Рисунок 5.7 – Макет мобільного дизайну головної сторінки

# **ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ**

## **6.1 Операційна система**

Для того, щоб людина могла користуватися web-додатком «Test system», на її девайсі повинен бути встановлений браузер (веб-оглядач).

Найпоширеніші браузери:

1. Google Chrome;
2. Firefox;
3. Safari;
4. Opera.

Це дозволить гарантувати якісний досвід роботи із web-додатком для більшості користувачів. Розробка програмного продукту спеціально під ці браузери дозволить врахувати всі особливості середовища та гарантувати якість продукту.

## **6.2 Мова програмування і середовище розробки**

В якості мови програмування серверної частини була вибрана мова JavaScript. Серверна частина була реалізована на програмній платформі NodeJs.

В якості мов програмування клієнтської частини були вибрані HTML5, SCSS, TypeScript. Клієнтська частина була написана з допомогою фреймворку Angular.

Angular може приймати рішення. Протягом всього процесу розробки доводиться приймати масу рішень - починаючи від характеристик продукту і до технологічних аспектів, які в майбутньому вплинуть на оперативність бази коду. Angular може приймати рішення і пропонує розробникам параметри за замовчуванням для таких речей як мережеве з'єднання, адміністративне управління станами, вибір мови, набір інструментів конфігурації.

Angular створений для Scale. Angular був створений в Google, щоб вирішити проблеми Google-scale. Для Google це означає мільйони рядків коду, тисячі інженерів, різноманітні графіки виконання проекту, вимоги і потік дій. Платформа була розроблена так, щоб дати можливість створювати і управляти загальним кодом, а також розділяти роботу між відповідними ролями.

Зручність супроводу - це наступна турбота багатьох розробників. Angular вирішує цю задачу двома шляхами. По-перше, розробники Angular використовують TypeScript, що дозволяє їм швидко знаходити баги і помилки. Крім цього, TypeScript дає можливість розробникам, які знайомі з базою коду, швидше включитися в роботу, так як вони зможуть швидше ознайомитися з типами даних в додатку. Команда Angular створила Angular з TypeScript саме тому. І нарешті, Angular робить ставку на зручність тестування. Введення залежностей - це ключовий елемент Angular, який полегшує завдання написання тестів. Angular включає і підтримку наскрізного тестування за допомогою Protractor.

Angular – надійний. У зв'язку з тим, що Angular - це продукт Google, він використовує всі переваги його інфраструктури тестування. Кожна зміна в Angular перевіряється на відповідність з кожним проектом Angular в межах Google. Це означає, що ще до будь-якого публічного релізу фреймворк вже використовують сотні проектів, а такий підхід збільшує шанси на відсутність непередбачених критичних змін або регрессий.

Angular має потужну підтримку. В інтернеті існує безліч багаторазово використовуваних інструментів, бібліотек і прикладів коду для Angular і AngularJS, і велика кількість цих інструментів або були оптимізовані під роботу з Angular, або зараз знаходяться в процесі оптимізації. Такі розробники як VMWare, Teradata, ag-Grid, NativeScript і інші вже повністю підтримують Angular.

## **6.3 Необхідні вимоги до конфігурації ЕОМ**

Мінімальні вимоги технічних засобів до клієнтської частини:

* операційна система: будь-яка;
* процесор: будь-який;
* браузер: будь-який.

Мінімальні вимоги технічних засобів до серверної частини:

* операційна система: будь-яка;
* процесор: будь-який;
* ОЗУ: 1ГБ.

# **ТЕОРЕТИЧНІ ВІДОМОСТІ**

## **7.1 Hypertext Markup Language**

HTML — стандартна [мова розмітки](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) в [Інтернеті](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Більшість [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) створюються за допомогою мови HTML (або [XHTML](https://uk.wikipedia.org/wiki/XHTML)). Документ HTML оброблюється [браузером](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від [SGML](https://uk.wikipedia.org/wiki/SGML), успадкувавши від неї [визначення типу документа](https://uk.wikipedia.org/wiki/Document_Type_Definition) та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є [мовою програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).

HTML разом із [каскадними таблицями стилів](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%BA%D0%B0%D0%B4%D0%BD%D1%96_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D1%96_%D1%81%D1%82%D0%B8%D0%BB%D1%96%D0%B2) та [вбудованими скриптами](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) — це три основні технології побудови [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0).

[Тім Бернерс-Лі](https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D1%80%D0%BD%D0%B5%D1%80%D1%81-%D0%9B%D1%96_%D0%A2%D1%96%D0%BC) представив HTML в дослідницькому центрі [CERN](https://uk.wikipedia.org/wiki/CERN) в [Женеві](https://uk.wikipedia.org/wiki/%D0%96%D0%B5%D0%BD%D0%B5%D0%B2%D0%B0)

1989 року.

1 HTML (без номера версії, 3 листопада 1992): найперша версія, орієнтована лише на текст.

2 HTML (без номера версії, 30 квітня 1993): до тексту додано атрибути, які визначають курсивне або жирне написання літер, та зображення.

3 HTML+ (листопад 1993): заплановані доповнення, які потрапили до наступних версій, але ніколи не були відокремлені як HTML+.

4 HTML 2.0 (листопад 1995): визначена стандартом [RFC 1866](https://tools.ietf.org/html/rfc1866) версія з підтримкою форм. Статус цього стандарту вже «історичний», також визнані застарілими попередні версії.

5 HTML 3.0: версія, яка не зазнала поширення, оскільки разом із випуском браузера [Netscape Navigator](https://uk.wikipedia.org/wiki/Netscape_Navigator) версії 3, цей стандарт вже був застарілим.

6 HTML 3.2 (14 січня 1997): були додані численні можливості, такі як таблиці, обтікання текстом зображень, інтеграція [аплетів](https://uk.wikipedia.org/wiki/%D0%90%D0%BF%D0%BB%D0%B5%D1%82).

7 HTML 4.0 (18 грудня 1997): були додані [таблиці стилів](https://uk.wikipedia.org/wiki/CSS), скрипти та фрейми. Також відбулось розділення на Strict (суворе дотримання стандартів), Frameset (з підтримкою фреймів), Transitional (перехідний). 24 квітня 1998 було випущено виправлену версію цього стандарту.

8 HTML 4.01 (24 грудня 1999): заміна версії HTML 4.0, містить численні дрібні виправлення.

9 [HTML 5](https://uk.wikipedia.org/wiki/HTML_5) (Working Draft, 5 квітня 2008):[[11]](https://uk.wikipedia.org/wiki/HTML#cite_note-11) HTML 5 має новий словник, побудований на основі HTML 4.01 та XHTML 1.0. Також перероблена і розширена пов'язана з HTML специфікація DOM.

10 [XHTML](https://uk.wikipedia.org/wiki/XHTML) 1.0 (26 січня 2000): висловлення стандарту HTML 4.01 засобами [XML](https://uk.wikipedia.org/wiki/XML). [1 серпня](https://uk.wikipedia.org/wiki/1_%D1%81%D0%B5%D1%80%D0%BF%D0%BD%D1%8F) 2002 було випущено оновлену редакцію стандарту.

11 XHTML 1.1 (31 травня 2001): після того, як XHTML буде розділено на модулі, стандарт XHTML 1.1 визначатиме сувору версію, в якій не буде запроваджених HTML 4 можливостей Frameset та Transitional.

12 XHTML 2.0 (розробка припинена в 2010 році): ця версія вже не базується на HTML 4.01 і додає деякі нові теги. Буде завершено розділення між представленням та вмістом.

HTML використовує найповнішу кодову таблицю [UCS](https://uk.wikipedia.org/w/index.php?title=%D0%A3%D0%BD%D1%96%D0%B2%D0%B5%D1%80%D1%81%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%9D%D0%B0%D0%B1%D1%96%D1%80_%D0%A1%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D1%96%D0%B2&action=edit&redlink=1) ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Universal Character Set — Універсальний Набір Символів).

Проте однієї кодової таблиці недостатньо для того, щоб браузери могли правильно відтворювати документи HTML. Для цього браузерам потрібно «знати» специфічну кодову таблицю документа, яку автор має зазначати завжди в елементі meta із параметром charset. Розмітка в HTML складається з чотирьох основних компонентів: елементів (та їхніх атрибутів), базових типів даних, символьних мнемонік та декларації типу документа.

Документ HTML 4.01 складається з трьох частин.

1 Декларація типу документа ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Document type declaration, Doctype), на початку документа, в якій визначається тип документа ([DTD](https://uk.wikipedia.org/wiki/Document_Type_Definition)).

2 Шапка документа (знаходиться в межах елемента head), в якій записано загальні технічні відомості або додаткова інформація про документ, яка не відтворюється безпосередньо в [браузері](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80);

3 Тіло документа (може знаходитися в елементах body або frameset), в якому міститься основна інформація документа.

Деякі елементи, наприклад br, не містять контенту, тож і не мають кінцевого тега. Елемент може не мати початкового та кінцевого тега (наприклад, елемент head), проте він завжди буде представлений в документі. Нижче зазначені деякі типи елементів розмітки HTML.

Більшість з атрибутів елемента являє собою пару «назва-значення», розділених між собою знаком рівняння, та записаних у початковому тегу одразу після назви елемента. Значення атрибуту може бути окреслено лапками (подвійними або одиничними), також, якщо значення атрибуту складається з певних символів, його можна не виділяти лапками зліва. Проте невзяття значення атрибутів у лапки вважається небезпечним кодом. На відміну від атрибутів виду «назва-значення», є певні атрибути, що впливають на елемент, назва яких лише з'явилась в початковому тегу (наприклад, атрибут ismap елемента img).

## **7.2 Cascading Style Sheets**

CSS — спеціальна [мова](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), що використовується для опису зовнішнього вигляду сторінок, написаних [мовами розмітки даних](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).

Найчастіше CSS використовують для візуальної презентації сторінок, написаних [HTML](https://uk.wikipedia.org/wiki/HTML) та [XHTML](https://uk.wikipedia.org/wiki/XHTML), але формат CSS може застосовуватися до інших видів [XML](https://uk.wikipedia.org/wiki/XML)-документів.

Специфікації CSS були створені та розвиваються [Консорціумом Всесвітньої мережі](https://uk.wikipedia.org/wiki/W3C).

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або [інтерфейсів](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81). Наприклад, існують профілі CSS для [принтерів](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80), мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0). Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), щоб визначити кольори, [шрифти](https://uk.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), [верстку](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B8) та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай [HTML](https://uk.wikipedia.org/wiki/HTML), [XML](https://uk.wikipedia.org/wiki/XML) або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані [монітора](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80), мобільного пристрою ([КПК](https://uk.wikipedia.org/wiki/%D0%9A%D0%B8%D1%88%D0%B5%D0%BD%D1%8C%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80)), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою [шрифту Брайля](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D0%B9%D0%BB%D1%8F_%D1%88%D1%80%D0%B8%D1%84%D1%82) або голосових браузерах та ін.).

Один і той самий [HTML](https://uk.wikipedia.org/wiki/HTML) або [XML](https://uk.wikipedia.org/wiki/XML) документ може бути відображений по-різному залежно від використаного CSS. Стилі для відображення сторінки можуть бути:

1. стилі автора (інформація надана автором сторінки);
2. стилі користувача;
3. стилі переглядача ([браузера](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80)).

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

Переваги :

1. інформація про стиль для усього [сайту](https://uk.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
2. різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на [принтер](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80), стиль для мобільних пристроїв;
3. сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;
4. прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

CSS має порівняно простий [синтаксис](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81) і використовує небагато англійських слів для найменування різних складових стилю.

Стилі складаються зі списку правил. Кожне правило має один або більше селектор та блок визначення. Блок визначення складається із оточеного фігурними дужками списку властивостей.

Властивості в списку оформлюються у вигляді назва властивості, двокрапка (:), значення, крапка з комою (;).

Властивість .class може мати не лише порівняння, але й перелік. Коли властивість class містить перелік значень, розділених пробілом, то до цього елементу застосовуються правила усіх перелічених класів. Наприклад, до <p class="note warning"> будуть застосовані правила note та warning.

Селектори можуть поєднуватися й іншим чином для досягнення більшої гнучкості. В наступному прикладі кілька селекторів об'єднуються через кому

Таблиці стилів в тому чи іншому вигляді існували з зародження [SGML](https://uk.wikipedia.org/wiki/SGML) в [1970-тих](https://uk.wikipedia.org/wiki/1970-%D1%82%D1%96). Каскадні таблиці стилів розроблялися для спрощення процедури додавання інформації про стилі для веб-сторінок.

З розвитком [HTML](https://uk.wikipedia.org/wiki/HTML) з'явилася можливість використовувати різні механізми для додавання стилів до елементів сторінки. Еволюція HTML дала веб-дизайнерам більше можливостей для створення вигляду сайту, але HTML-код ставав складнішим для написання та зміни. Через різницю у відображення сторінки в різних браузерах збереження стилю сторінки було складним, користувачі мали менше контролю над відображенням контенту.

На розгляд W3C було запропоновано дев'ять різних варіантів таблиць стилів. Після обговорення в спеціальному списку розсилки було обрано два, вони створили основу для того, що стало CSS.

В 1994 році було створено World Wide Web Consortium [W3C](https://uk.wikipedia.org/wiki/W3C), серед інших питань W3C займався також і CSS. Робочу групу очолив [Steven Pemberton](https://uk.wikipedia.org/w/index.php?title=Steven_Pemberton&action=edit&redlink=1), як провідні технічні спеціалісти до неї входили [Håkon Wium Lie](https://uk.wikipedia.org/w/index.php?title=H%C3%A5kon_Wium_Lie&action=edit&redlink=1) та [Bert Bos](https://uk.wikipedia.org/w/index.php?title=Bert_Bos&action=edit&redlink=1).

В [грудні](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%B4%D0%B5%D0%BD%D1%8C) [1996](https://uk.wikipedia.org/wiki/1996) було опубліковано CSS рівня 1 — [CSS level 1 Recommendation](http://www.w3.org/TR/CSS1). В квітні 2016, опубліковано CSS рівня 2 - [Cascading Style Sheets Level 2 Revision 2 (CSS 2.2)](https://www.w3.org/TR/CSS22/), робота над яким ще продовжується.

## **7.3 Angular**

Angular (зазвичай так називають фреймворк Angular 2 або Angular 2+, тобто вищі версії) — написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій. Angular — це AngularJS, який переосмислили та який був повністю переписаний тією ж командою розробників.

Спочатку переписаний AngularJS отримав назву Angular 2 від команди розробників, яка над ним працювала, але це призвело до плутанини серед інших розробників. Аби пояснити різницю між ними та наголосити, що це окремі проекти, команда вирішила для фреймворків версій 1.X застосовувати назву AngularJS, а для версій, починаючи з 2.0, — Angular без JS.

Основні відмінності між Angular та AngularJS

Архітектура додатка на Angular. Основними елементами в розробці є модулі, компоненти, шаблони, метадані, біндінг даних, директиви, сервіси та ін'єкції залежностей.

Як згадано вище, Angular — це ретельно переписаний AngularJS.

* додано Angular CLI, що дає змогу розпочати створення нового додатка, просто написавши команду ng new [app name]
* Angular не використовує концепцію "області видимості" або контролерів, натомість як головну архітектурну концепцію він застосовує ієрархію компонентів
* Angular має інакший синтаксис написання виразів, застосовуючи "[ ]" для біндінгу даних властивостей, і "( )" для біндінгу даних івентів
* модульність – значна частина основного функціоналу перенесена у модулі
* Angular рекомендує та застосовує розроблену Microsoft мову — TypeScript, що містить такі можливості, як:
* класи, а отже Об'єктно-орієнтоване програмування
* система типізації
* узагальнене програмування
* TypeScript — надмножина ECMAScript 6 (ES6), і є зворотно сумісним зі стандартом ECMAScript 5 (тобто JavaScript).Angular також має такі ES6-можливості, як:
* анонімні функції
* ітератори
* цикли типу For/Of
* Python-подібні генератори
* рефлексія
* динамічне завантаження
* асинхронна компіляція шаблонів
* заміна контролерів та $scope(області видимості) компонентами та директивами – компонент є директивою з шаблоном
* ітеративні колбеки завдяки використанню RxJS. RxJS дещо обмежує видимість станів та можливості дебагінгу, але, застосовуючи такі плагіни, як ngReact та ngrx, це легко вирішується.

## **7.4 TypeScript**

TypeScript — мова програмування, представлена Microsoft восени 2012; позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript.

Розробником мови TypeScript є Андерс Гейлсберг (англ. Anders Hejlsberg), який створив раніше C#, Turbo Pascal і Delphi.

Код експериментального компілятора, котрий транслює код TypeScript в представлення JavaScript, поширюється під ліцензією Apache, розробка ведеться в публічному репозиторії через сервіс CodePlex. Специфікації мови відкриті і опубліковані в рамках угоди Open Web Foundation Specification Agreement (OWFa 1.0).

TypeScript є зворотньо сумісним з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js.

Переваги над JavaScript:

* можливість явного визначення типів (статична типізація),
* підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
* підтримка підключення модулів.

За задумом ці нововведення мають підвищити швидкість розробки, прочитність, рефакторинг і повторне використання коду, здійснювати пошук помилок на етапі розробки та компіляції, а також швидкодію програм.

Планується, що в силу повної зворотної сумісності адаптація існуючих застосунків на нову мову програмування може відбуватися поетапно, шляхом поступового визначення типів. Підтримка динамічної типізації зберігається — компілятор TypeScript успішно обробить і не модифікований код на JavaScript. Основний принцип мови — весь існуючий код на JavaScript сумісний з TypeScript, тобто в програмах на TypeScript можна використовувати стандартні JavaScript-бібліотеки і раніше створені напрацювання. Більш того, можна залишити існуючі JavaScript-проекти в незмінному вигляді, а дані про типізації розмістити у вигляді анотацій, які можна помістити в окремі файли, які не заважатимуть розробці і прямому використанню проекту (наприклад, подібний підхід зручний при розробці JavaScript-бібліотек).

На момент релізу представлені файли для сприйняття розширеного синтаксису TypeScript для Vim і Emacs, а також плагін для Microsoft Visual Studio.

Одночасно з виходом специфікації розробники підготували файли з деклараціями статичних типів для деяких популярних JavaScript-бібліотек, серед яких jQuery.

# **ВИСНОВКИ**

Web-додатки є актуальними в наш час. Це технологія, яка стрімко розвивається і покликана зекономити час та ресурси на створенні універсальних додатків (програм), які в свою чергу дають можливість працювати в будь-якому місці, у будь-який час та з будь-яких девайсів не залежно від операційної системи чи технічних характеристик.

В даній курсовій роботі був розроблений web-додаток «Test system». Результати розробки наведені у додатку A. Для досягнення поставленої цілі було виконано технічне завдання.

До недоліків роботи можна віднести наступне:

* + не реалізована можливість реєстрації різних користувачів;
  + не реалізована можливість виставлення балів.

За допомогою цього додатку можна проводити попереднє тестування кандидатів на посади в компанії, зменшуючи таким чином фінансові витрати та витрати часу на даний процес.

При оформленні курсової роботи були одержані навики написання програмного забезпечення, а також великий практичний досвід роботи з:

* Angular;
* HTML5 ;
* CSS/SCSS ;
* JavaScript;
* TypeScript.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1 angular.io [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://angular.io/ (дата звернення 10.10.2018). – Angular.

2 developer.mozilla.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://developer.mozilla.org/ru/docs/Learn/CSS3 (дата звернення 11.10.2018). – css.

3 webref.ru [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://webref.ru/css (дата звернення 11.10.2018). – html.

4 developer.mozilla.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://developer.mozilla.org/ru/docs/Learn/HTML (дата звернення 11.10.2018). – html.

5 webref.ru [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://webref.ru/html (дата звернення 11.10.2018). – html.

6 www.muicss.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://www.muicss.com/ (дата звернення 11.10.2018). – html.

7 sass-lang.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://sass-lang.com/ (дата звернення 27.10.2018). – SCSS.

8 www.typescriptlang.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: http://www.typescriptlang.org/ (дата звернення 01.11.2018). – TypeScript.

9 stackoverflow.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані.– Режим доступу: https://stackoverflow.com/ (дата звернення 01.11.2018). – Angular.