

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ

ГЕНЕРАЦІЇ РЕЗЮМЕ

SOFTWARE DEVELOPMENT FOR RESUME

GENERATING

Виконав: студент(ка) 4 курсу, групи КНТ-228

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Сташук Д.А.

(прізвище та ініціали)

Керівник Федорончак Т.В.

(прізвище та ініціали)

Рецензент Терещенко Е.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки і технології
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Сташука Дениса Андрійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для генерації резюме.

Software Development for Resume Generating

керівник проєкту (роботи) Федорончак Тетяна Василівна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Опис програми. 3. Розробка програмного забезпечення. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту(роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Федорончак Т.В., доцент		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “ 04 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту(роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1-2 тижні	Розділ 1
3	Опис програми.	3 тиждень	Розділ 2
4	Розробка програмного забезпечення.	3-4 тижні	Розділ 3
5	Експлуатація, тестування та експериментальне	4 тиждень	Розділ 4
6	дослідження програми.		
7	Оформлення пояснювальної записки та документів до неї.	5 тиждень	Додатки
8	Нормоконтроль та рецензування.	6 тиждень	
9	Захист роботи.	7 тиждень	

Студент(ка)

(підпис)

Сташук Д.А.

(прізвище та ініціали)

Керівник проєкту(роботи)

(підпис)

Федорончак Т.В.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 117 с., 69 рис., 2 табл., 3 дод., 18 джерел.

ГЕНЕРАЦІЯ РЕЗЮМЕ, ВЕБЗАСТОСУНОК, HTML, БІБЛІОТЕКА JQUERY, МОБА JAVASCRIPT.

Об'єкт дослідження – програмні засоби для створення резюме.

Предмет дослідження – вебзастосунок, що надає користувачам можливість створювати резюме.

Мета роботи – програмна реалізація вебзастосунку для створення та завантаження резюме.

Матеріали, методи та технічні засоби: середовище розробки Microsoft Visual Studio Code, основні інструменти для веброзробки HTML/CSS, мова програмування JavaScript, набір функцій jQuery.

Результати. Розроблено програмний вебзастосунок для створення та генерації резюме.

Висновки. Результати роботи повністю відповідають поставленій меті та визначеному технічному завданню до проєкту. Проведено аналіз та порівняння з існуючими подібними застосунками. Розроблено програмний продукт, який призначений для створення резюме.

Галузь використання – проєкт може бути використаний для створення персональних резюме.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 117 pages, 69 figure, 2 tables, 3 appendixes, 18 sources.

CV GENERATION, WEB APPLICATION, HTML, JQUERY LIBRARY, JAVASCRIPT LANGUAGE.

The object of the research is the software tools for creating a resume.

The subject of the research is a web-application, which allows users to create resumes.

The purpose of the work is the program realization of the web application for creation and loading of resume.

Materials, methods and technical tools: Microsoft Visual Studio Code development environment, basic tools for HTML/CSS web development, JavaScript programming languages, jQuery feature set.

Results. Developed a web-based application for creating and generating resumes.

Conclusions. The results of the work fully correspond to the set goal and defined technical task to the project. Analysis and comparison with existing similar applications. A software product designed to create a resume has been developed.

Field of use – the project can be used to create personal resume.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	10
1.1 Опис предметної області	10
1.2 Аналіз аналогів	11
1.2.1 «Cvmaker.ua»	11
1.2.2 «Jobbit.com»	12
1.2.3 «Canva.com»	13
1.2.4 Порівняльна таблиця існуючих програмних продуктів	14
1.3 Постановка завдання роботи	15
1.4 Висновки до розділу 1	16
2 Опис програми	17
2.1 Основні засоби розробки	17
2.1.1 Мова гіпертекстової розмітки HTML	17
2.1.2 Каскадна таблиця стилів CSS	18
2.1.3 Bootstrap	19
2.1.4 Мова програмування JavaScript	20
2.1.5 Набір функцій jQuery	22
2.2 Вибір середовища розробки	24
2.3 Схема взаємодії компонентів ПЗ	24
2.4 Вимоги до програмного забезпечення	25
2.5 Аналіз вимог до розробки дизайну програми	27
2.6 Висновки до розділу 2	28
3 Розробка програмного забезпечення	29
3.1 Розробка дизайну вебзастосунку	29
3.2 Опис модулів реалізованого вебзастосунку	41
3.3 Структурна схема проєкту	42
3.4 Опис реалізованих функцій	43

3.5 Висновки до розділу 3	45
4 Експлуатація, тестування та експериментальне дослідження програми	46
4.1 Призначення програми.....	46
4.2 Умови виконання програми	46
4.3 Тестування в критичних середовищах	47
4.4 Тестування програми	47
4.5 Висновки до розділу 4.....	63
Висновки.....	64
Перелік джерел посилання	66
Додаток А Технічне завдання.....	68
Додаток Б Текст програми.....	72
Додаток В Слайди презентації	107

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- AJAX – Asynchronous JavaScript and XML;
- API – Application Programming Interface;
- BOM – Browser Object Model;
- CV – Curriculum Vitae;
- DOM – Document Object Model;
- HDD – Hard Disk Drive;
- PDF – Portable Document Format;
- SSD – Solid-State Drive;
- UI – User Interface;
- XML – eXtensible Markup Language;
- ОС – операційна система;
- ПЗ – програмне забезпечення.

ВСТУП

Опис своїх навичок при працевлаштуванні за свою історію сильно змінився. Спочатку при прийомі на роботу резюме взагалі не існувало. Потім з'явилася автобіографія, де автор у стислій формі описував себе та свої навички. І вже на сьогодні існує така чудова річ як резюме.

Створення вебзастосунку для генерації резюме має багато переваг, серед наведених вище методів опису себе як спеціаліста. Наприклад, такий вебзастосунок відображає всі необхідні пункти, які повинен заповнити користувач та має вже готовий для використання шаблон.

Актуальність та проблематика роботи полягає в тому, що з часом резюме стало невід'ємною частиною працевлаштування. Тому запит суспільства на створення такого вебзастосунку для генерації резюме є великим. Таким чином, заповнювати резюме стає простіше, маючи тільки свої особисті дані, а про все інше потурбується вебзастосунок. Також це значно зменшить час та зусилля людини для створення власного резюме.

Метою дипломної кваліфікаційної роботи проекту є оцінка та аналіз існуючих методів та програмна реалізація вебзастосунку для генерації резюме.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Резюме – документ, який містить в собі інформацію про навички, досвід роботи, освіти та іншу інформацію, яку зазвичай вимагають для розгляду кандидатури людини для найму.

Складання резюме – нелегке завдання, особливо якщо хочеться, щоб воно було професійним, сучасним і, як результат, успішним.

На сьогоднішній день одним з головних критеріїв для працевлаштування є якісне резюме. Грамотно складене резюме – перша умова успішного отримання місця в компанії. Воно є одним з найефективніших засобів самореклами на ринку праці.

Основні вимоги до написання резюме:

- інформацію потрібно викладати чітко, лаконічно, грамотно та правдиво. Неможна використовувати специфічні символи та аббревіатури;
- вказати інформацію, яку ви можете підтвердити та описати тільки позитивні якості.

Резюме зазвичай складається з кількох розділів, інформація у кожному з яких покликана у вигляді ознайомити потенційного роботодавця з усіма відомостями про кандидата, необхідними для його прийому на роботу. У резюме не слід включати інформацію, яка не має жодного відношення до передбачуваної роботи. У резюме слід включати лише інформацію, яка допоможе максимально охарактеризувати вас.

Резюме повинно містити:

- коротку інформацію про кандидата;
- короткий опис посади, на яку він претендує;
- коротку характеристику основних навичок;
- опис досвіду роботи по спеціальності у зворотному хронологічному порядку (зазвичай, три останні місяця роботи);
- опис освіти (дипломи, сертифікати);

- доступні рекомендації та досягнення.

Іноді також включають інформацію про членство у професійних асоціаціях, наукових публікаціях та авторських патентах. Також у резюме можна включити інформацію про ваші захоплення, краще якщо це активні захоплення (наприклад, спорт, туризм, танці), однак такий опис має бути викладено коротко, і не варто включати в резюме марну інформацію.

У різних країнах структура резюме або CV може відрізнятися. Останнім часом набули широкого поширення Інтернет-резюме, що заповнюються за певним шаблоном і розміщуються на спеціалізованих сайтах рекрутингових агенцій. Існують також професійні служби, що пропонують складання грамотних резюме [1].

Ще одним, поки що малопоширеним, типом резюме є відео-резюме, яке є коротким відеороликом, в якому кандидат розповідає про себе, свої навички та ін. І так само, як і у випадку з Інтернет-резюме, для них існують спеціалізовані сайти, на яких можна їх розмістити, та компанії, які займаються їх записом та розміщенням.

1.2 Аналіз аналогів

В наш час з'явилися онлайн рішення для складання резюме. Це ніщо інше, як конструктори або генератори резюме. Вони мають чітку структуру та допомагають скоротити час для складання резюме.

1.2.1 «Cvmaker.ua»

Cvmaker.ua [2] – вебзастосунок для створення резюме. Він має зручний та зрозумілий інтерфейс. З його допомогою всього за пару кроків можна створити резюме. Після створення резюме його можна завантажити у форматі PDF. На рис. 1.1 наведено першу сторінку створення резюме за допомогою «Cvmaker.ua».

The image shows a web form for creating a resume on the cvmaker website. The form is titled 'Особисті дані' (Personal Data) and is part of a three-step process: 'Особисті дані', 'Досвід', and 'Шаблон'. The form includes a 'Додати фото' (Add photo) button, a language dropdown set to 'Українська', and several input fields: 'Ім'я*' (Name), 'Прізвище*' (Surname), 'Адреса електронної пошти*' (Email), 'Номер телефону' (Phone), 'Адреса' (Address), 'Поштовий індекс' (Postal code), and 'Місто' (City) with an example 'наприклад, Лондон'. A 'Додаткова інформація' (Additional information) button is at the bottom.

Рисунок 1.1 – Перша сторінка створення резюме

Переваги «Cvmaker.ua»:

- адаптивність. Даний вебзастосунок можливо використовувати як на персональному комп'ютері, так і на смартфоні;
- зовнішня взаємодія. Вебзастосунок виконує всі необхідні функції. Можливість створення резюме та подальше завантаження.

Вебзастосунок для створення резюме володіє важливою перевагою – з ним зручно складати опис майбутнього кандидата.

Головним недоліком цього ресурсу є платність. Безкоштовно завантажити резюме не вдасться.

1.2.2 «Jobbit.com»

Наступним аналогом є «Jobbit.com» [3]. Відносно молодий сервіс, в якому з'явилася можливість створення резюме. Цей ресурс дещо відрізняється від попереднього. Структура створення резюме знаходиться на одній сторінці, що є доволі зручним для використання. На рис. 1.2 наведено користувацький інтерфейс вебзастосунку «Jobbit.com».

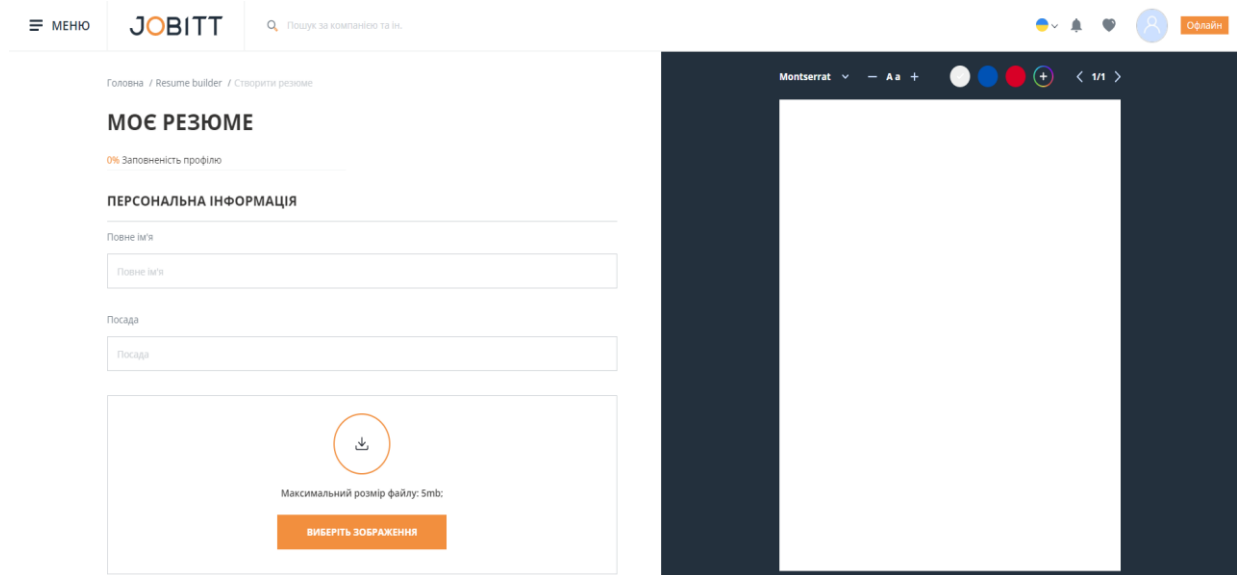


Рисунок 1.2 – Користувацький інтерфейс сайту «Jobbit.com»

Переваги «Jobbit.com»:

- адаптивність. Розглядаємий сервіс можливо використовувати як на персональному комп'ютері, так і на смартфоні;
- прогрес створення резюме у відсотках, що в свою чергу дає розуміння загального обсягу резюме;
- відображення введених даних у реальному часі в окремому вікні, яке знаходиться на самій сторінці створення резюме;
- безкоштовне завантаження створеного резюме.

1.2.3 «Canva.com»

Ще одним аналогом є «Canva.com» [4]. Canva дозволяє ефективно працювати, спілкуватися з колегами та підтримувати єдність фірмового стилю абсолютно всім: і любителям роботи поодиночі, і учасникам команд – маленьких, великих, регіональних чи міжнародних. Цей ресурс відрізняється від попередніх. Користувач вводить дані безпосередньо на самому шаблоні, що має велику перевагу над розглянутими вище застосунками. На рис. 1.3 наведено користувацький інтерфейс вебзастосунку «Canva.com».

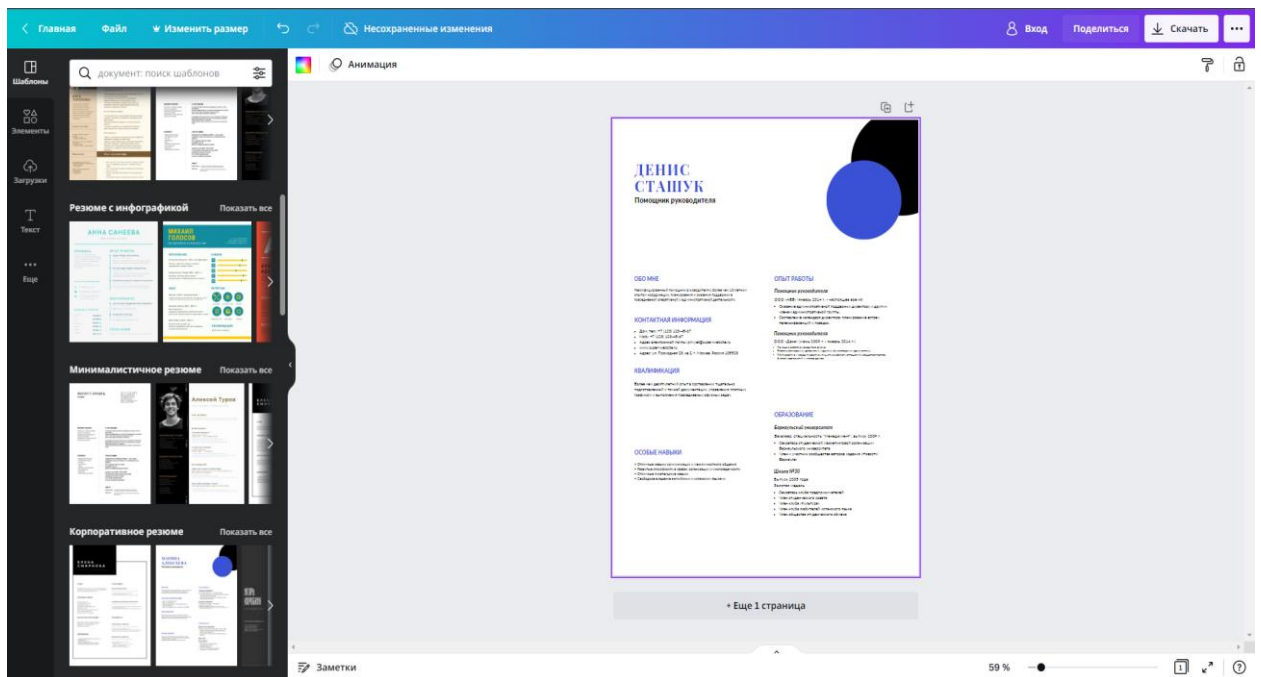


Рисунок 1.3 – Користувачський інтерфейс сайту «Canva.com»

Переваги «Canva.com»:

- адаптивність. Сервіс можливо використовувати на персональному комп'ютері, смартфоні або планшеті;
- введення даних на самому шаблоні, що робить процес створення резюме простішим та зрозумілішим, адже результат відображається миттєво;
- безкоштовне завантаження створеного резюме.

1.2.4 Порівняльна таблиця існуючих програмних продуктів

Вище було наведено одні з багатьох рішень, які дозволяють створити резюме онлайн. Але, звичайно, крім переваг, усі вони мають і недоліки.

Серед характерних переваг можна виділити наступні функції для певних сервісів: вони мають простий та зручний інтерфейс, основні функції, безкоштовність, легкість використання, доступність.

Серед недоліків можна виділити не зовсім зручний інтерфейс деяких з них, платність та відсутність адаптивності.

В табл. 1.1 проведено порівняння вже існуючих рішень.

Таблиця 1.1 — Порівняння програмних продуктів

	«Cvmaker.ua»	«Jobbit.com»	«Canva.com»
Виконання основного функціоналу (створення резюме, його завантаження)	+	+	+
Адаптивність	+	+	-
Зручний інтерфейс	+	+	-
Платність	+	-	-
Підтримка української мови	+	+	+
Швидкодія	-	+	+
Перегляд резюме у реальному часі	-	+	-

1.3 Постановка завдання роботи

Оскільки завданням роботи є реалізація вебзастосунку для генерації резюме, можна виділити такі завдання на виконання:

- проаналізувати інформаційне забезпечення;
- розробити інтерфейс;
- обрати мову програмування та середовище розробки;
- виконати програмну реалізацію;
- провести тестування та виправлення помилок.

1.4 Висновки до розділу 1

В даному розділі було досліджено предметну область. В ході дослідження було виявлено, що існує велика кількість вебзастосунків для створення резюме. На базі цього аналізу було сформовано список переваг, які варто враховувати при створенні власного застосунку, а саме, безкоштовність, зручний та простий інтерфейс, виконання основних функцій, швидкодія, адаптивність.

2 ОПИС ПРОГРАМИ

2.1 Основні засоби розробки

2.1.1 Мова гіпертекстової розмітки HTML

HTML (HyperText Markup Language — «мова гіпертекстової розмітки») — базовий будівельний блок Інтернету. Він визначає зміст та структуру вебконтенту. Інші технології, крім HTML, зазвичай використовуються для опису зовнішнього вигляду/подання (CSS) або функціональності/поведінки (JavaScript) вебсторінки.

Під гіпертекстом ("hypertext") розуміються посилання, які з'єднують вебсторінки один з одним або в межах одного вебсайту або між вебсайтами. Посилання є фундаментальним аспектом Інтернету. Завантажуючи контент в Інтернет і пов'язуючи його зі сторінками, створеними іншими людьми, ви стаєте активним учасником Всесвітньої мережі [5].

Зазвичай середній вебсайт містить кілька різних HTML-сторінок. Наприклад: домашні сторінки, звичайні сторінки, сторінки контактів матимуть окремі HTML-документи.

Кожна HTML-сторінка складається з набору тегів (також званих елементами), які можна назвати будівельними блоками вебсторінок. Вони створюють ієрархію, яка структурує контент за розділами, параграфами, заголовками та іншими блоками контенту.

Більшість елементів HTML мають відкриття та закриття, в яких використовується синтаксис `<tag></tag>`.

HTML-теги мають два основних типи: блок-рівень та вбудовані теги.

Елементи рівня блоку займають весь вільний простір і завжди запускають новий рядок у документі. Заголовки та параграфи – відмінний приклад блокових тегів.

Вбудовані елементи займають стільки місця, скільки їм потрібно і не запускають новий рядок на сторінці. Вони зазвичай служать форматування

внутрішнього вмісту елементів рівня блока. Посилання та підкреслені рядки – гарні приклади вбудованих тегів.

HTML4 (в наші дні зазвичай званий HTML) був опублікований в 1999 році, а остання велика версія вийшла в 2014 році. HTML5 — це оновлення, яке ввело багато нових функцій.

Однією з найбільш очікуваних особливостей HTML5 є підтримка вбудовування аудіо та відео. Замість використання Flash-плеєра ми можемо просто вставляти відео та аудіофайли на наші вебсторінки за допомогою нових тегів `<audio>` `</audio>` та `<video>` `</video>`. Він також включає вбудовану підтримку масштабованої векторної графіки (SVG) та MathML для математичних та наукових формул.

HTML5 також запровадив кілька семантичних вдосконалень. Нові семантичні теги інформують браузер про значення контенту, що приносить користь як читачам, так і пошуковим системам.

2.1.2 Каскадна таблиця стилів-CSS

Cascading Style Sheets (CSS) - це мова ієрархічних правил (таблиць стилів), що використовується для представлення зовнішнього вигляду документа, написаного на HTML або XML (включаючи різні мови XML, такі як SVG та XHTML). CSS описує, як елемент повинен відобразитися на екрані, на папері, голосом або за допомогою інших засобів масової інформації.

CSS є однією з основних мов вільної веброзробки, яка стандартизована специфікацією W3C. Стандарт CSS поділяється на рівні: CSS1 в даний час застарів, CSS2.1 - рекомендований для застосування, а CSS3, розбитий на дрібніші модулі, розвивається на шляху стандартизації [6].

CSS був розроблений W3C (World Wide Web Consortium) у 1996 році з досить простої причини. У HTML не були розроблені теги, які б допомогли відформатувати сторінку. Потрібно було лише написати розмітку для сайту.

Такі теги, як ``, були введені в HTML версії 3.2 і це створило багато проблем для розробників. Оскільки вебсайти мали різні шрифти, кольоровий фон та стилі, переписувати код було довгим, болючим та дорогим процесом. Таким чином, CSS був створений W3C для вирішення цієї проблеми.

Зовнішній метод може бути найзручнішим. Все робиться зовні у файлі `.css`. Це означає, що ви можете робити всі стилі в окремому файлі та застосовувати CSS до будь-якої сторінки, яку ви хочете. Зовнішній стиль також може покращити час завантаження.

Вбудований стиль CSS працює з конкретними елементами, що мають тег `<style>`. Кожен компонент повинен бути стилізований, тому він може бути не найкращим або швидким способом обробки CSS. Але це може стати в нагоді. Наприклад, якщо ви бажаєте змінити один елемент, швидко переглянути зміни або, можливо, у вас немає доступу до файлів CSS.

2.1.3 Bootstrap

Bootstrap – це відкритий та безкоштовний HTML, CSS та JS фреймворк, який використовується веброзробниками для швидкої верстки адаптивних дизайнів сайтів та вебдодатків.

Bootstrap складається з:

- інструментів для створення макета (обгорткових контейнерів, потужної системи сіток, гнучких медіа-об'єктів, адаптивних утилітних класів);
- класів для стилізації базового контенту: тексту, зображень, коду, таблиць та `figure`;
- готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, слайдерів, списків, акордеонів, модальних вікон, спливаючих підказок та ін;
- утилітних класів для вирішення традиційних завдань, що найбільш часто виникають перед веброзробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, `margin` і `padding`

відступів, і т.д. [7].

2.1.4 Мова програмування JavaScript

JavaScript (часто просто JS) – це легковажна, інтерпретована або JIT-компілювана, об'єктно-орієнтована мова з функціями першого класу. Найширше застосування знаходить мову сценаріїв вебсторінок, але також використовується і в інших програмних продуктах, наприклад, node.js або Apache CouchDB. JavaScript це прототипно-орієнтована, мультипарадигменна мова з динамічною типізацією, яка підтримує об'єктно-орієнтовану, імперативну та декларативну (наприклад, функціональне програмування) стилі програмування [8].

Код JavaScript зазвичай або вбудований у вашу сторінку, або посилається на окремий файл .js. Це мова, яка працює на стороні клієнта, що означає, що скрипт завантажується на комп'ютер відвідувача і обробляється там, на противагу тому, як працюють сторони сервера (виконуються на сервері до того, як відправити файл відвідувачам).

Слід враховувати, що більшість браузерів також пропонують можливість користувачам відключати JavaScript. Тому слід продумати, що станеться у випадку, якщо на якомусь комп'ютері код не інтерпретуватиметься.

Для розробки вебзастосунку доцільно використати мову програмування JavaScript.

JavaScript – мультипарадигменна мова програмування. Була створена Бренданом Ейхом. Підтримує об'єктно-орієнтований, імперативний та функціональний стилі. Являє собою реалізацію специфікації ECMAScript.

JavaScript використовується як вбудована мова для програмного доступу до об'єктів програм. Найширше застосування знаходить у браузерах як мова сценаріїв надання інтерактивності вебсторінкам.

Основні архітектурні риси: динамічна типізація, слаба типізація, автоматичне керування пам'яттю, прототипне програмування, функції, як об'єкти першого класу.

Переваги JavaScript:

- незамінність для веброзробки (підтримка скриптів всіма популярними браузерами, повна інтеграція з версткою сторінок та серверною частиною);
- швидкість роботи і продуктивність (частково дозволяє обробляти вебсторінки на комп'ютерах користувача без запитів до сервера, що в результаті економить час та трафік, а також знижує навантаження на сам сервер);
- потужна інфраструктура (безліч прикладів, що в свою чергу робить процес розробки легким та приємним);
- зручність користувацьких інтерфейсів (заповнення форм, вибір дій, активація кнопок, перевірки введення даних, реакція на наведення курсора) [9].

Недоліки JavaScript:

- відсутність можливості зчитування та завантаження файлів (обмеження функціональності на стороні клієнта – підтримання безпечності);
- нестрога типізація (різна інтерпретація даних, відсутність можливості раннього виявлення помилок);
- доступність для шахраїв (у вільну скриптову мову програмування найпростіше вбудувати фрагмент шкідливого коду).

Структура мови програмування:

- ядро (ECMAScript);
- об'єктна модель браузера (BOM);
- об'єктна модель документа (DOM).

На сьогодні підтримку JavaScript забезпечують сучасні версії найчастіше використовуваних браузерів. В Internet Explorer, Opera, Mozilla

Firefox, Safari, Google Chrome наявна повна підтримка третьої редакції ECMA-262.

На думку творця даної мови програмування, підтримка в Internet Explorer компанією Microsoft одного із існуючих та використовуваних в інших браузерах швидких движків JavaScript може призвести до появи додатків, які працюють з тривимірною графікою, написаних на JavaScript 3D-ігор, використанні JavaScript в задачах, де раніше застосовувалась технологія Adobe Flash.

2.1.5 Набір функцій jQuery

jQuery - це швидка, маленька та багата на функції бібліотека JavaScript. Вона робить речі, такі як HTML-обробка документів і маніпуляції, обробка подій, анімацій і Ajax набагато простішими з простим у використанні API, що працює в безлічі браузерів [10].

Бібліотека jQuery містить такі можливості [11]:

- маніпуляція HTML/DOM;
- маніпуляція CSS;
- методи подій HTML;
- ефекти і анімації;
- AJAX;
- утиліти.

jQuery робить роботу з Ajax набагато простіше. Ajax працює асинхронно з рештою коду. Це означає, що код, написаний на Ajax, може взаємодіяти з сервером та оновлювати його вміст без необхідності перезавантаження сторінки.

Однак це пов'язано із проблемами. Різні браузери виконують Ajax API по-різному. Таким чином, код повинен відповідати всім браузерам. Вручну, це важка та трудомістка робота. На щастя, jQuery виконує всю важку роботу та адаптує код для всіх браузерів.

jQuery – це JavaScript-бібліотека, яка фокусується на взаємодії JavaScript, HTML и CSS. Її створив Джон Резіг.

Що вміє jQuery:

- не тільки звертатися до будь-якого елемента DOM, але і маніпулювати ним;
- працювати з подіями;
- легко реалізовувати різноманітні візуальні ефекти;
- працювати з AJAX;
- має велику кількість JavaScript-плагінів, які використовуються для створення елементів користувацьких інтерфейсів.

Роботу з jQuery можна розділити на 2 типи:

- отримання jQuery-об'єкта за допомогою функції `$()`. Якщо метод не повинен нічого повертати, то він повертає посилання на jQuery-об'єкт, що в результаті дозволяє створювати ланцюжок вікликів методів.
- виклик глобальних методів у об'єкта `$`.

Переваги jQuery:

- оптимізація сучасного коду під устарілі браузері;
- прискорення роботи з JavaScript;
- використання сучасних плагінів;
- швидка відладка, тестування додатка.

Недоліки jQuery:

- відсутність рівня даних;
- не підходить для складних користувацьких інтерфейсів;
- великий об'єм файлу для імпорту;
- застарілий API [12].

Отже, враховуючи перераховані особливості та переваги, для розробки вебзастосунку було обрано мову програмування JavaScript, крім того, також було обрано набір функцій jQuery для створення певних візуальних ефектів.

2.2 Вибір середовища розробки

Visual Studio Code – це легкий, але потужний редактор вихідного коду, який працює на вашому робочому столі та доступний для Windows, macOS та Linux. Він поставляється із вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов (таких як C++, C#, Java, Python, PHP, Go) та середовищ виконання (таких як .NET та Unity) [13].

Visual Studio Code має високопродуктивний редактор коду, який у поєднанні із службами мов програмування дає вам можливості IDE та швидкість текстового редактора. Провідник відмінно підходить для навігації між файлами при вивченні проєкту. VS Code надає потужні команди для навігації по файлах за допомогою простих у використанні клавіш [14].

Visual Studio Code має інтегроване управління системою контролю версій (SCM) та включає готову підтримку Git . Багато інших постачальників системи управління версіями доступні через розширення на VS Code Marketplace [15].

JavaScript IntelliSense у Visual Studio Code забезпечує інтелектуальне завершення коду, інформацію про параметри, пошук посилань та багато інших розширених функцій мови. JavaScript IntelliSense заснований на службі JavaScript, розробленій командою TypeScript. Хоча IntelliSense має працювати для більшості проєктів JavaScript без будь-яких налаштувань, можна зробити IntelliSense ще більш корисним за допомогою JSDoc або шляхом налаштування jsconfig.json проєкту [16].

2.3 Схема взаємодії компонентів ПЗ

Для написання певного функціоналу вебзастосунку було обрано мову програмування JavaScript та набір функцій jQuery.

Загальна схема взаємодії компонентів програмного забезпечення зображена на рис. 2.1.

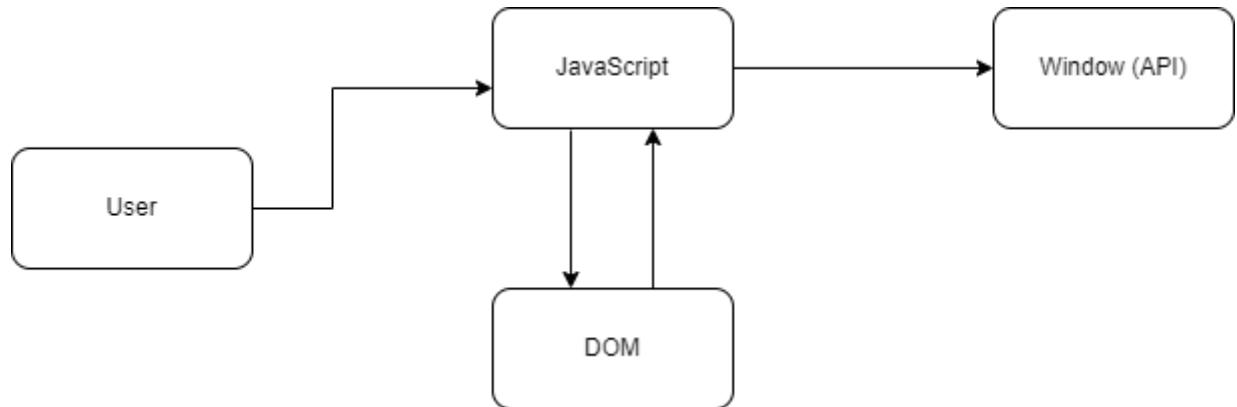


Рисунок 2.1 – Загальна схема взаємодії компонентів ПЗ

2.4 Вимоги до програмного забезпечення

Програмний продукт повинен забезпечувати виконання таких функцій:

- завантаження особистого зображення;
- введення персональних даних;
- введення контактної інформації;
- супровідний лист;
- інформація про попередні місця роботи;
- перелік закінчених освітніх закладів чи курсів;
- інформація про особисті професійні навички;
- введення інформації для зворотнього зв'язку;
- завантаження резюме.

Схема функцій користувача представлена на рис. 2.2.

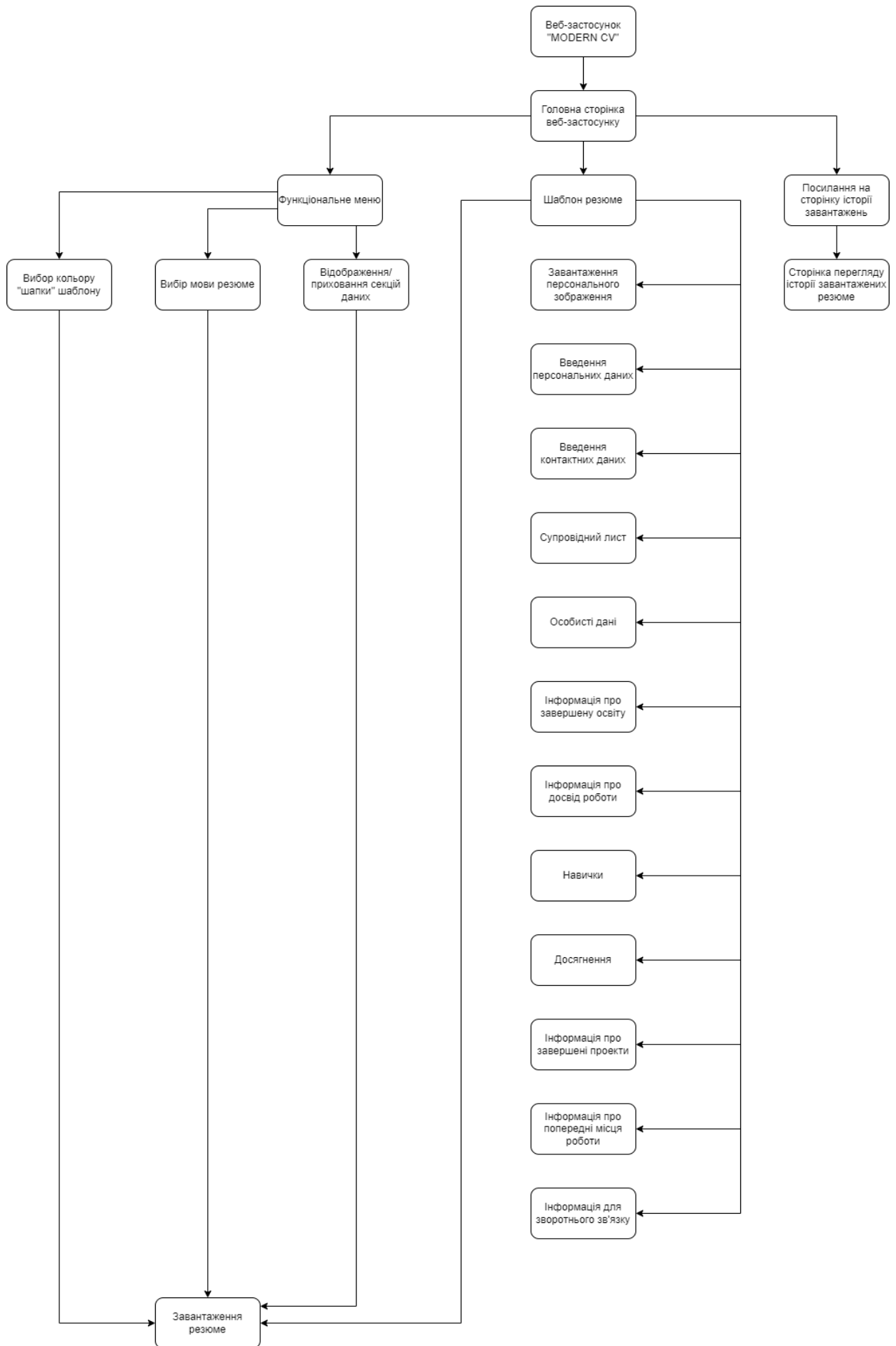


Рисунок 2.2 – Схема функцій користувача

На рис. 2.3 наведено діаграму прецедентів для роботи з вебзастосунком.

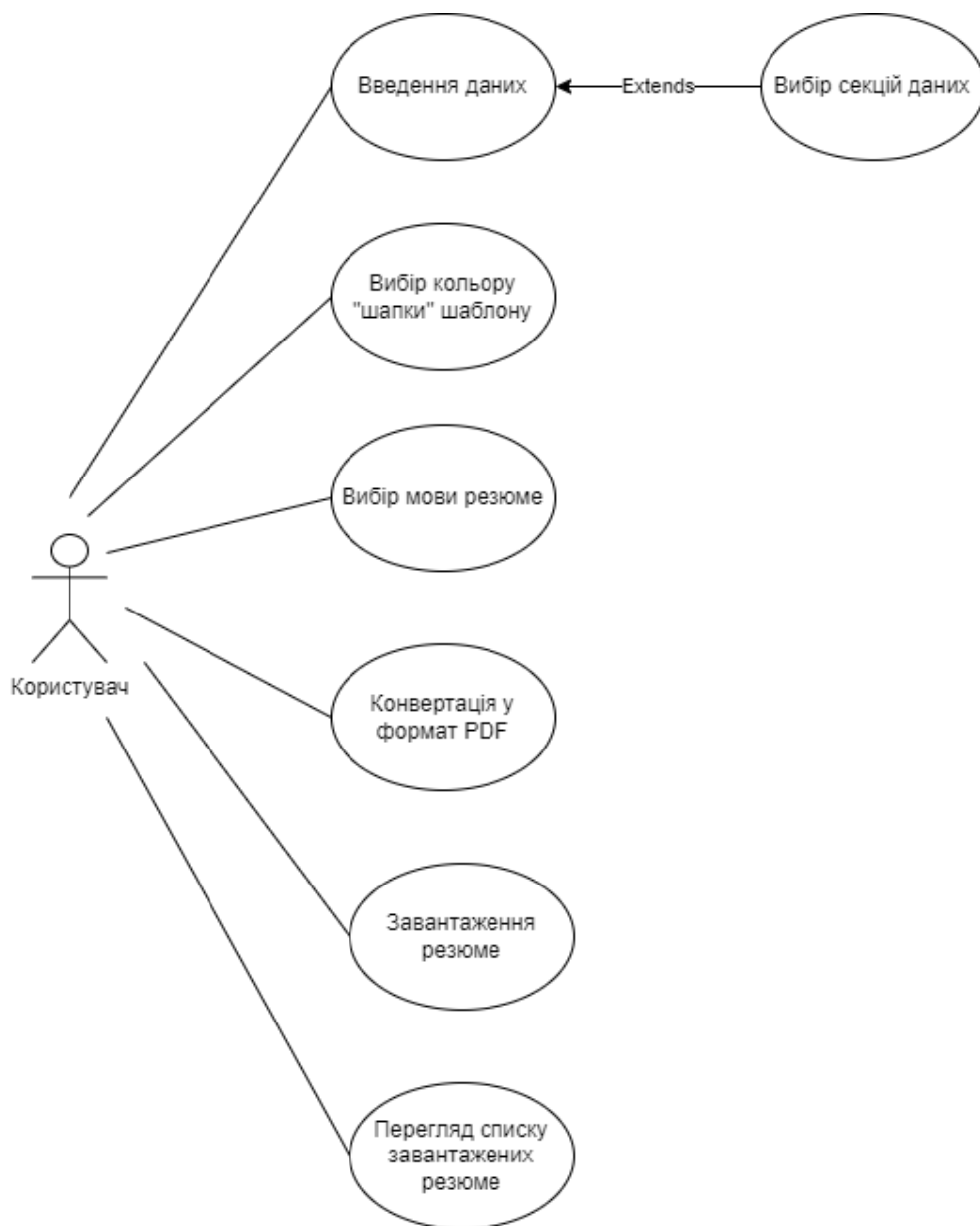


Рисунок 2.3 – Діаграма прецедентів користувача

2.5 Аналіз вимог до розробки дизайну програми

Оскільки розроблювана програма призначена для створення резюме, то основний акцент потрібно робити в аналітичному стилі, а саме в конкретній секції потрібно розміщувати важливу інформацію для швидкого візуального аналізу, який не займе багато часу. Тому в результаті дизайн вебзастосунку

було створено у мінімалістичному стилі, а саме з максимальним дотриманням його принципів.

Мінімалізм – певний стиль в дизайні, який характеризується лаконічністю виразних засобів, простотою, точністю і якістю композиції. Мінімалісти не враховують класичні прийоми творчості та традиційні художні матеріали. Вони використовують природні та промислові матеріали простих геометричних форм, нейтральних кольорів і малих обсягів.

На сьогодні мінімалізм є дуже популярним. Він має безліч переваг: користувач чітко розуміє де саме він знаходиться на сторінці і що йому потрібно робити. Багато полів, зайвих кольорів та зображень призводять до того, що увагою користувачів легко керувати. Мінімалістичний стиль дає змогу робити менше помилок, що в результаті сприяє до швидкої роботи. Окремі елементи та загальний дизайн в цілому добре запам'ятовуються [17].

Основним фактором мінімалізму у дизайні є простий та зрозумілий інтерфейс.

Головний принцип в мінімалізмі при створенні дизайну – не використовувати багато кольорів. Зазвичай, використовують лише декілька. Він також використовується в створенні іконок, картинок.

2.6 Висновки до розділу 2

В цьому розділі було розглянуто основні інструменти для розробки вебзастосунку, такі як HTML, CSS, Bootstrap, JavaScript, jQuery.

Було представлено загальну архітектуру програмного забезпечення, сформульовано функціональні вимоги до розроблюваного застосунку та розглянуто стиль дизайну вебзастосунку.

Також був наведений короткий опис текстового редактору Visual Studio Code. Цей редактор має багато переваг, однією з яких є безкоштовність. З набором додаткових плагінів він перетворюється в потужний інструмент для виконання різноманітних задач.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка дизайну вебзастосунку

Вебдизайн — вид графічного дизайну, спрямований на розробку та оформлення об'єктів інформаційного середовища інтернету, що забезпечує їм високі споживчі властивості та естетичні якості. Подібне трактування відокремлює вебдизайн від вебпрограмування, підкреслює специфіку предметної діяльності вебдизайнера, позиціонує вебдизайн як графічний дизайн.

Вебзастосунок — це програма, яка працює через браузер користувача за аналогією з мобільним, при цьому підключення до Інтернету необов'язкове.

Переваги:

- щоб запустити вебзастосунок користувачеві, не потрібно встановлювати додаткові програми: він запускається з інтернетом на будь-якій операційній системі;
- у процесі розробки не потрібно малювати окремий дизайн для Windows, Linux чи Mac OS;
- простий процес оновлення – користувачу не потрібно нічого завантажувати та встановлювати.

Оскільки для створення програмного інтерфейсу використовується мінімалістичний стиль, тому основними кольорами шаблону було обрано один з відтінків блакитного (#0064ff) та білий (ffffff). Для бекграунду вебзастосунку використовувався градієнт з наступних кольорів: бірюзовий (#1ac4db), смарагдовий (#00c495) та жовто-зелений (#8ccc14).

Було створено бургер-меню з певними функціями для покращення гнучкості функціоналу розроблюваного ПЗ. З його допомогою користувачу надається можливість зміни кольору «хідера», вибір мови, якою буде створено резюме та відображення або приховання певних блоків шаблону (реалізується за допомогою DOM-маніпуляцій). Результат зображено на рис. 3.1.

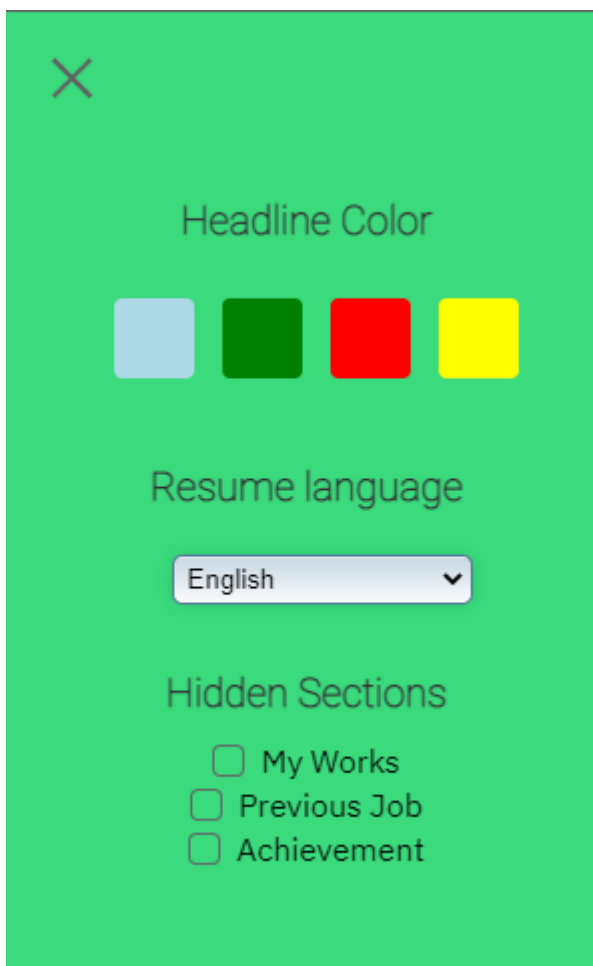


Рисунок 3.1 – Функціональне бургер-меню

Також було розроблено окрему сторінку для перегляду історії завантажених резюме. Посилання на цю сторінку знаходиться у верхній частині вебзастосунку (рис. 3.2).

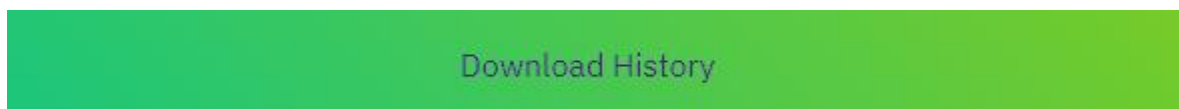


Рисунок 3.2 – Посилання на сторінку історії завантажених резюме

Сторінка історії завантажених резюме має наступний вигляд (рис. 3.3).



Рисунок 3.3 – Сторінка історії завантажених резюме

За замовчуванням, у блакитному кольорі виконано верхню частину шаблону (header). Результат зображено на рис. 3.4.



Рисунок 3.4 – Header

Як вказувалося раніше, а допомогою функціоналу реалізованого бургер-меню можна змінювати колір «шапки» шаблону резюме. Для цього спочатку отримуємо id «шапки» за допомогою DOM, а потім встановлюємо відповідний колір (рис. 3.5).

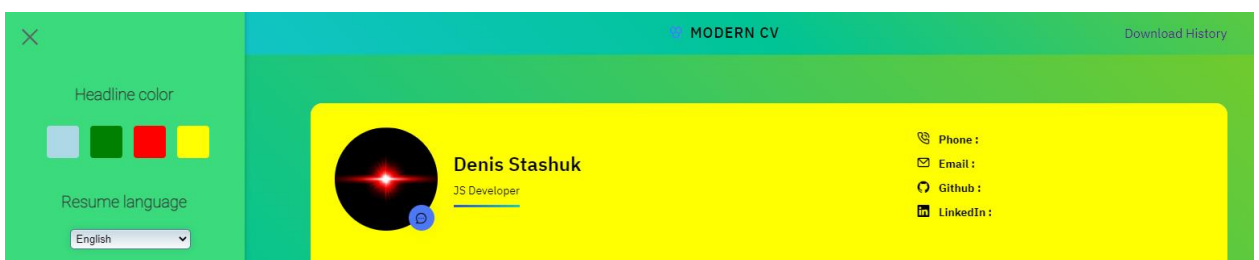


Рисунок 3.5 – Зміна кольору «шапки» шаблону

Крім того, при наведенні курсору миші на верхню частину шаблону з'являється анімація «ефекту води». Це додає загальному вигляду інтерфейсу вебзастосунку додаткової візуалізації та заспокійливим чином впливає на користувача, заспокоюючи нервову систему, знижуючи рівень стресу та просто є приємним доповненням (рис. 3.6).



Рисунок 3.6 – «Ефект води»

Основна частина шаблону виконана у білому кольорі (рис. 3.7).

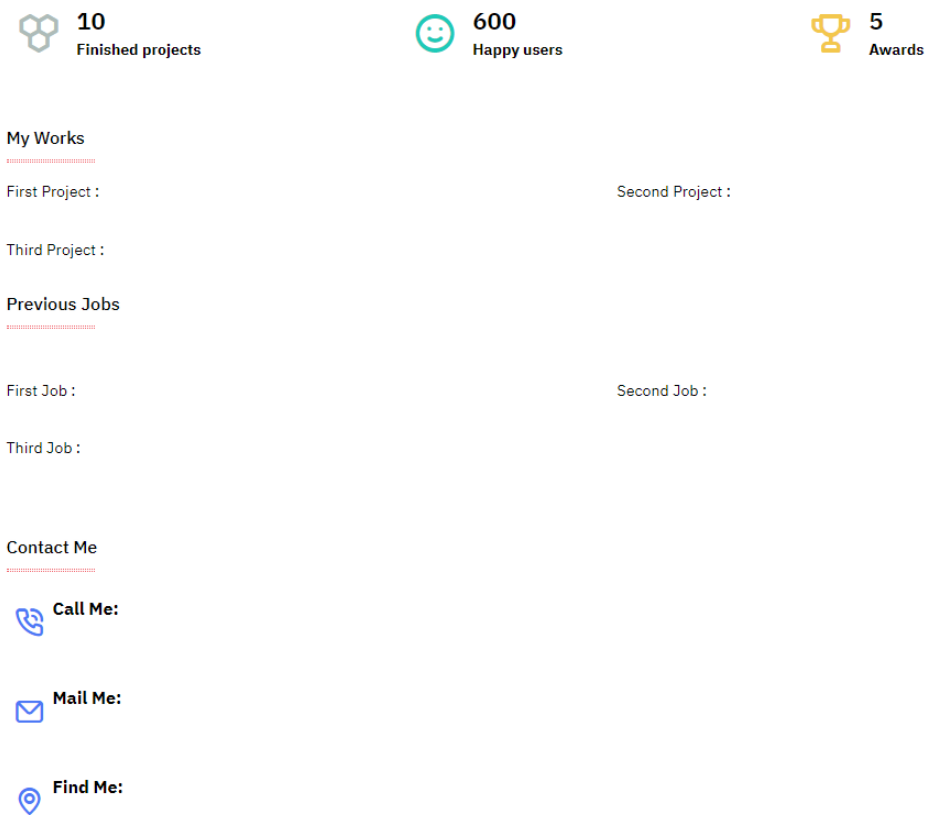


Рисунок 3.7 – Приклад основної частини шаблону

Однією з функцій користувача є можливість змінювати мову, якою буде написано резюме (рис. 3.8).

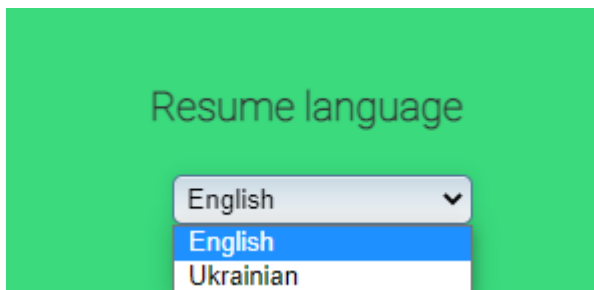


Рисунок 3.8 – Меню зміни мови резюме

Результат зміни мови написання резюме показано на рис. 3.9 та 3.10.

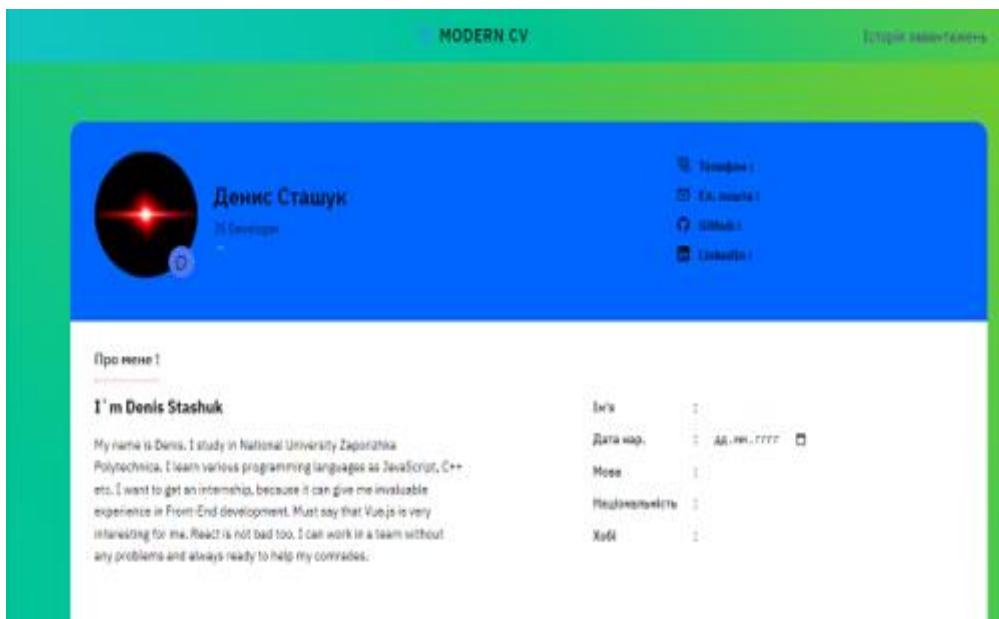


Рисунок 3.9 – Результат зміни мови написання резюме

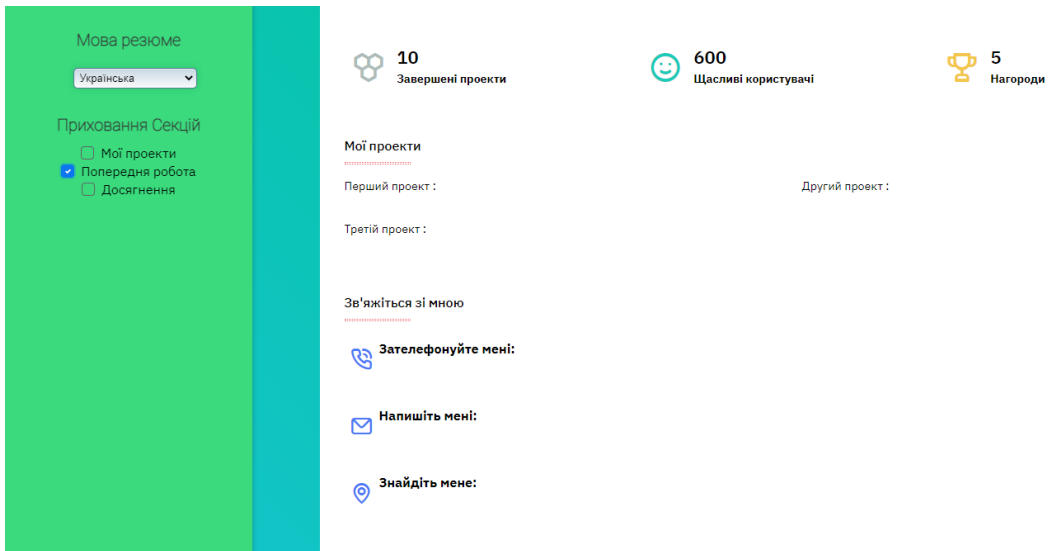


Рисунок 3.12 – Приховання секції «Попередня робота»



Рисунок 3.13 – Приховання секції «Досягнення»

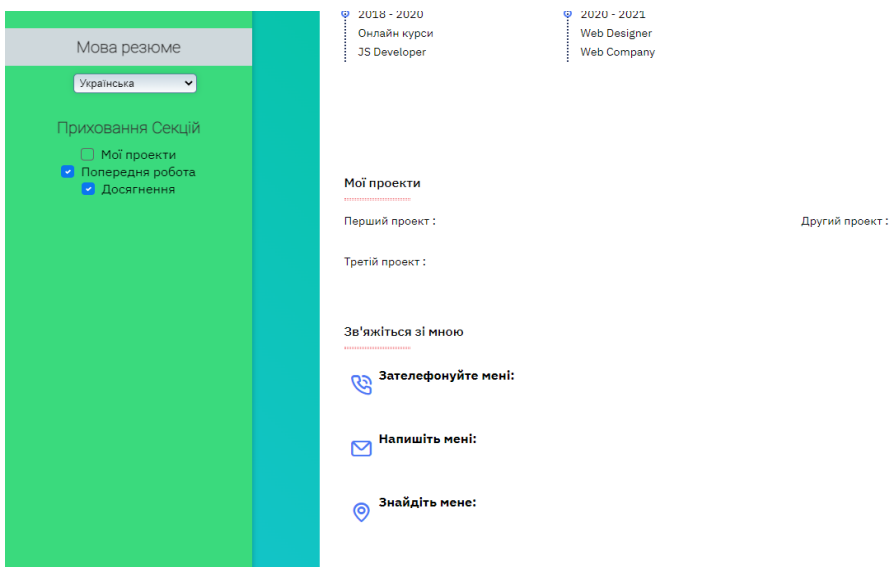


Рисунок 3.14 – Приховання секцій «Попередня робота» та «Досягнення»

Виділення заголовків деяких секцій здійснювалось за допомогою підкреслення відтінком червоного кольору (#f1646c). Результат показано на рис. 3.15.

Contact Me



Рисунок 3.15 – Виділення заголовку однієї з секцій

Поєднання блакитного та білого кольорів дає чудовий результат. Але потім виникає питання щодо кольору, який потрібно застосувати у вигляді фону самої сторінки вебзастосунку. Після проведення багатьох дослідів з цією метою було вирішено обрати градієнт трьох кольорів.

Внизу сторінки вебзастосунку для створення резюме знаходиться кнопка завантаження (рис. 3.16).

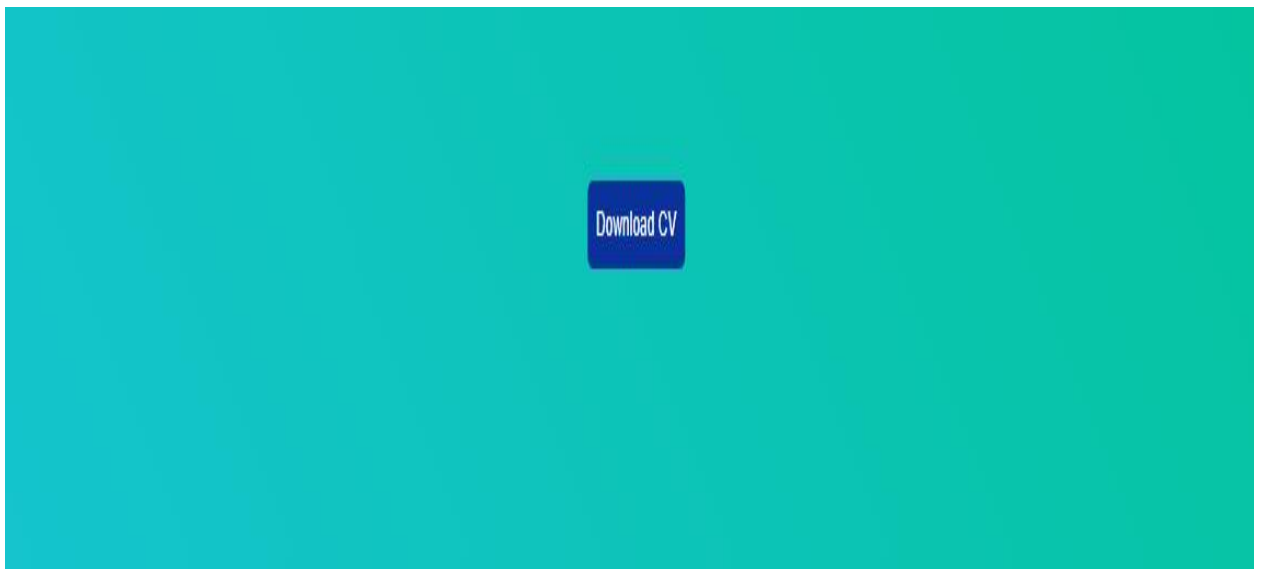


Рисунок 3.16 – Кнопка завантаження резюме

Загальний вигляд сторінки створення резюме показано на рис. 3.17.

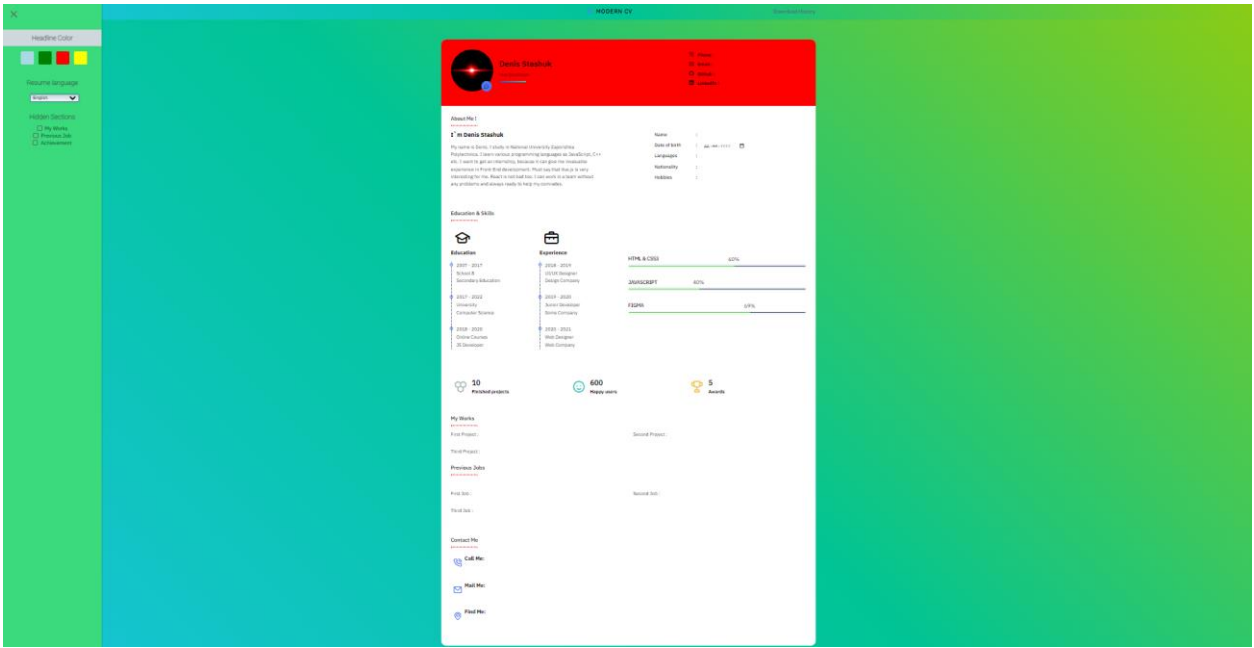


Рисунок 3.17 – Загальний вигляд сторінки створення резюме

Варто звернути увагу, що для заповнення шаблону було використано англійську мову.

Чорним кольором (#000000) було позначено шрифт, а сімейство шрифтів було обрано IBM Plex Sans. Якщо проаналізувати цей шрифт, то можна виділити такі аспекти: лаконічність, мінімалізм та легкість в розпізнанні (рис. 3.18).

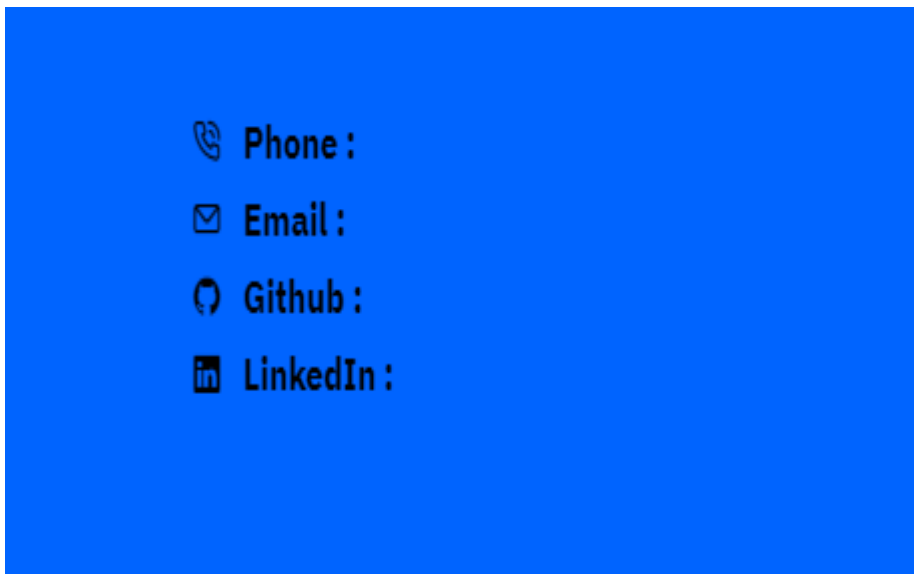


Рисунок 3.18 – Використання шрифту IBM Plex Sans

Значок, також іконка (від англ. icon) - елемент графічного інтерфейсу, невелика картинка, що позначає додаток, файл, каталог, вікно, компонент операційної системи, пристрій.

Як правило, іконки є растровими або векторними картинками квадратного формату стандартних розмірів.

В операційних системах Microsoft Windows найбільш уживані розміри іконок зі стороною 16, 32, 48 та 256 пікселів. У різних середовищах також використовуються іконки з розміром 24, 40, 60, 72, 92, 108, 128 пікселів та інші. Зі збільшенням розширення моніторів сучасні операційні системи включають підтримку все більших розмірів іконок: до 256x256 для Windows 10, до 512x512 для macOS.

Зазвичай файл іконки містить кілька окремих зображень різних розмірів і з різною кількістю використовуваних кольорів. Це з тим, що з автоматизованому зменшенні растрового зображення воно втрачає різкість, а при збільшенні — деталізацію. Це особливо помітно, коли зменшення або збільшення піддаються зображення маленького розміру, характерні для іконок. Тому трудомісткість виготовлення повного комплекту високоякісних зображень для растрової іконки є досить високою. Ця проблема вирішується під час використання векторних іконок (наприклад, у форматі SVG). Використання векторних іконок поки що неможливо у Windows або Mac OS X, але вже підтримується операційною системою Linux.

Багато сучасних середовищ і програм вміють ставити іконки за замовчуванням. Іконка програми вибирається стандартною, для файлу — залежно від типу файлу (часто береться іконка програми, що обробляє цей файл за замовчуванням). Іноді іконки для графічних файлів виходять із вмісту самого файлу (наприклад, зменшена копія зображення, англ. thumbnail). Деякі просунуті інтерфейси можуть робити те й для неграфічних файлів (документів, презентацій). Не всі іконки статичні. Наприклад, іконка, що представляє зв'язок з Інтернетом, може наочно показувати, чи йде передача даних, іконка диспетчера завдань — показувати завантаження процесора і т.д.

Вебсайти також можуть мати іконки, що відображаються в адресному рядку та закладках браузера. Такі іконки виготовляються за технологією favicon.

Векторна графіка міцно увійшла до життя сучасної людини. Рекламні плакати, візитки, сайти. Це далеко не повний список об'єктів, на яких зустрічаються векторні зображення.

Векторна графіка – це один із способів представлення графічних об'єктів і зображень в комп'ютерній графіці, заснований на математичному описі елементарних геометричних об'єктів, таких як: точки, лінії, сплайни, криві Безьє, кола, еліпси, багатокутники.

Об'єкти векторної графіки описуються сукупністю координат, параметрів та атрибутів.

Термін "векторна графіка" використовується для підкреслення відмінностей від растрової графіки, в якій зображення представлене у вигляді графічної матриці, що складається з пікселів.

При виведенні на матричні пристрої відображення векторна графіка попередньо перетворюється на растрову графіку. Перетворення проводиться програмно або апаратно за допомогою сучасних відеокарт. Для плотерів і векторних моніторів перетворення з векторного в растровий формат не потрібне, тому що в таких пристроях примітиви будуються переміщенням пера або електронного променя.

Цей тип графіки є сукупністю математично описаних об'єктів. Цим обумовлені необмежені можливості масштабування. При збільшенні растрового зображення обов'язково губиться його якість, обсяг векторних ілюстрацій навпаки можна вільно змінювати, не втрачаючи їх візуальних параметрів. Це можливо завдяки тому, що кожна точка растрової картини має свої координати, а між точками розташовані криві, зовнішній вигляд яких залежить від розташування точок.

Ще однією перевагою векторних зображень є їхня невелика вага. Такі файли можна легко надіслати електронною поштою, а з їх редагуванням впорається будь-який комп'ютер.

Важливою перевагою векторних картинок є їхня чіткість при широкоформатному друку, що визначає їх популярність при створенні рекламних банерів.

Також безперечною перевагою векторної графіки є те, що дизайнер має можливість працювати з окремими фрагментами зображень. Є можливість переміщати та обертати окремі лінії та області, змінювати колір фрагментів картинки, групувати об'єкти тощо.

Отже, жоден сайт не може обійтися без іконок. Вони візуалізують певні розділи або якісь важливі поля. В даній роботі використовувались мінімалістичні векторні іконки Unicons [18]. Іконки мають приємний дизайн, який радує око. Результат показано на рис. 3.19.



Рисунок 3.19 – Використання іконок Unicons

Безпосередньо, сам шаблон має заокруглені кути. Прямокутники з заокругленими кутами легше сприймаються оком, ніж прямокутники з гострими краями, тому що вони потребують менше когнітивних зусиль для візуальної обробки жовтою плямою. Заокруглені кути направлені всередину і до центра прямокутника, що робить акцент на його змісті.

Скругленість виглядає сучасно. Застосування повного заокруглення почалося в мобільних інтерфейсах і перекочувало до Інтернету. Відсутність гостроти викликає почуття простоти, оптимізму та доступності. Ймовірно,

саме тому багато систем дизайну мають заокруглені елементи і широко застосовують їх для іконок, кнопок або навіть зображень.

У Хромі після недавнього апдейту адресний рядок став округленим, щоб краще виразити доступність одночасного використання для пошуку. Користувачі можуть побачити частину результатів пошуку прямо під час друку ще до переходу на сторінку.

Немає однозначної відповіді, що краще — заокругленість або гострокутність. Але заокруглені елементи можуть допомогти спонукати користувача взаємодіяти з програмою, не відволікаючись на другорядний UI.

3.2 Опис модулів реалізованого вебзастосунку

Вся інформація на сторінці вебзастосунку розділена на певні блоки, аби детальніше розділити інформацію та полегшити процес розробки. Всі секції знаходяться в тезі `<body>`:

- `<nav>` - блок, в якому знаходиться назва, логотип, посилання на сторінку з історією завантажених резюме та бургер-меню вебзастосунку;
- `<section class="profile" id="profile_ripple">` - секція, де знаходиться основна інформація (зображення, ім'я, номер телефону, електронна пошта, посилання на соціальні мережі);
- `<section class="about">` - дана секція відповідає за уточнюючу інформацію (дата народження, володіння іноземними мовами, національність, інтереси) та супровідний лист;
- `<section class="skills">` - в цій секції знаходиться інформація про здобуту освіти (3 поля для введення даних), досвід праці (3 поля для введення даних) та рівень володіння технологіями (3 поля для введення даних);
- `<section class="counter_section">` - секція відображає завершені проєкти, кількість задоволених користувачів та нагороди;
- `<section class="work">` - в даному блоці перераховуються завершені проєкти та інформація про попередні місця роботи;

- `<section class="contact">` - остання секція, яка бере на себе роль відображення інформації для зворотного зв'язку.

3.3 Структурна схема проєкту

В цьому підрозділі буде представлено структурну схему проєкту. Структура директорій проєкту представлена на рис. 3.20.

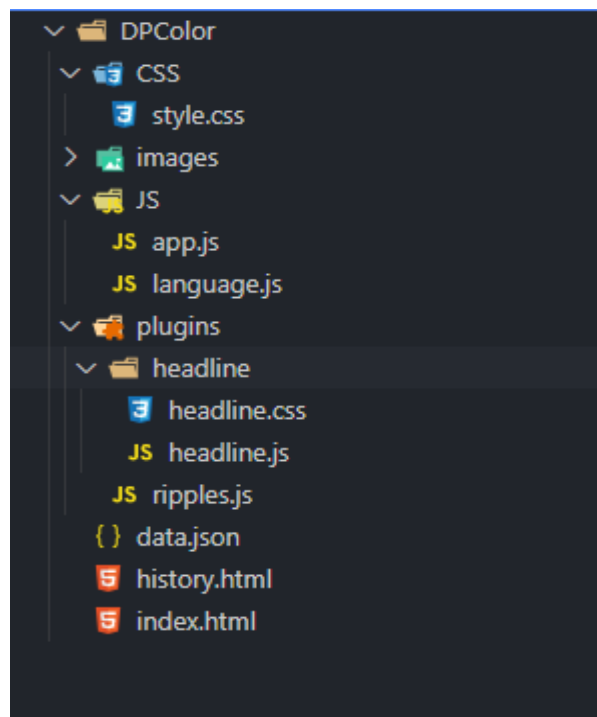


Рисунок 3.20 – Структура директорій проєкту

Директорія CSS призначена для зберігання файлу `style.css`, який відповідає за стилі шаблону вебзастосунку.

Директорія `images` містить в собі зображення для завантаження до шаблону (персональна фотографія) та логотип сторінки вебзастосунку.

Директорія JS призначена для зберігання файлу `app.js`, в якому знаходиться реалізація анімації рівня володіння технологіями та лічильників завершених проєктів, кількості задоволених користувачів та нагород. Також тут знаходиться файл `language.js`, який відповідає за зміну мови створення резюме.

Директорія plugins містить піддиректорію headline, яка в свою чергу містить файли headline.css та headline.js, котрі відповідають за стилі градієнту та анімацію самого градієнту в секції `<section class="profile" id="profile_ripple">`. Також директорія plugins містить файл ripples.js. Даний файл несе в собі реалізацію «ефекта води».

Директорія index.html призначена для зберігання основної структури проєкту та підключення інших файлів.

Директорія history.html відповідає за сторінку історії завантажених резюме. При зберіганні файлу резюме, до файлу data.json додається інформація про збережений файл, а саме дата та час збереження, адреса збереження, та поточний лічильник завантаженого файлу. Збереження такої інформації дозволяє зробити історію завантажених резюме, список яких можна подивитися на відповідній сторінці.

3.4 Опис реалізованих функцій

В даному підрозділі розглянемо деякі функції та змінні, які будуть використовуватись у проєкті. Розглянемо функції та змінні, що відповідають за реалізацію певного функціоналу та візуалізацію певних ефектів.

Функції та змінні файлу app.js:

- function() - головна (загальна) функція, яка містить в собі інші функції та змінні (звернення до документу за допомогою jQuery);

- const bars – змінна, яка зберігає в собі значення володіння тією чи іншою технологією у відсотках;

- const counters – змінна (лічильник), яка зберігає в собі дані з секції `<section class="counter_section">`;

- runCounter() – функція, яка активує лічильник (анімація від 0 до вказаного користувачем значення при прокрутці сторінки);

- printCV() – функція, яка відповідає за функціонал генерації резюме.

Функції та змінні файлу headline.js:

- var animationDelay – змінна, яка встановлює час затримки анімації;
- var barAnimationDelay – змінна, яка встановлює час затримки ефекта
полоси завантаження;
- var lettersDelay – змінна, яка встановлює час затримки виводу тексту;
- var revealAnimationDelay – змінна, яка встановлює затримку ефекта
«кліпу»;
- initHeadline() – функція, яка відповідає за вставку елемента для кожної
букви змінюємого слова;
- animateHeadline() – функція ініціалізації анімації заголовка;
- singleLetters() – функція, яка відповідає за заміну тексту при активації
анімації;
- setTimeout(function()) – функція запуску анімації;
- hideWord() – функція, яка приховує слово при появі наступного;
- showWord() – функція, яка показує наступне слово після того як
попереднє слово буде приховане;
- takeNext() – функція, яка показує перше слово після закінчення всього
переліку слів (якщо далі немає слів, то буде відображатися перше слово і так
по колу).
- loadJSON() – функція, яка відповідає за отримання даних про історію
збережень резюме з файлу data.json. Оскільки дані зберігаються у зовнішньому
файлі, дана функція робить XMLHttpRequest до цього файлу, отримує дані з нього
та додає їх на сторінку історії збережених резюме. Дані файлу data.json
зображені на рис. 3.21.

```

DPColor > {} data.json > ...
1  [
2  { "number": 1, "address": "D:\\Downloads\\ModermCVFinal.pdf", "datetime": "13.06.2022 13:49"},
3  { "number": 2, "address": "D:\\Data\\ModermCVFinal.pdf", "datetime": "13.06.2022 13:59"},
4  { "number": 3, "address": "D:\\Docs\\ModermCVFinal.pdf", "datetime": "13.06.2022 14:03"},
5  { "number": 4, "address": "D:\\Downloads\\ModermCVFinal(1).pdf", "datetime": "13.06.2022 14:10"},
6  { "number": 5, "address": "D:\\Downloads\\ModermCVFinal(2).pdf", "datetime": "13.06.2022 14:11"},
7  { "number": 6, "address": "D:\\Downloads\\ModermCVFinal(3).pdf", "datetime": "14.06.2022 11:16"},
8  { "number": 7, "address": "D:\\Downloads\\ModermCVFinal(4).pdf", "datetime": "14.06.2022 11:20"}
9  ]

```

Рисунок 3.21 – Дані файлу data.json

3.5 Висновки до розділу 3

В даному розділі було проаналізовано та обґрунтовано вибір мови програмування. Було розроблено дизайн вебзастосунку та розкрито переваги того чи іншого стилів. Описано модулі та структурну схему реалізованого проєкту. Наведено деякі функції та змінні, які використовуються у програмі.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення програми

Розроблювана програма призначена для створення та завантаження резюме, що в свою чергу допоможе ще на один крок приблизитися до успішного працевлаштування. Програмне забезпечення надає доступ до введення всієї необхідної інформації, а саме: персональні та контактні дані, супровідний лист, інтереси, рівень володіння тією чи іншою мовою, інформація про попередні місця роботи та розроблені проєкти та ще багато якої інформації.

Програма дозволяє швидко та зручно створити якісне, сучасне резюме без зайвих зусиль та насолодитися процесом створення.

4.2 Умови виконання програми

Для експлуатації розробленої програми необхідні наступні програмно-технічні засоби:

- комп'ютер з встановленою операційною системою Windows 10, 8.1 або 7;
- текстовий редактор коду Microsoft Visual Studio Code останньої версії;
- процесор – не менше 800МГц;
- об'єм оперативної пам'яті пристрою – не менше 4 Гб для повноцінної та комфортної роботи;
- монітор та відеокарта з роздільною здатністю не менше 1024x768, а також смартфон;
- обсяг вільного простору на HDD чи SSD не менше 1 Гб;
- підключення до мережі Інтернет.

4.3 Тестування в критичних середовищах

Метою даного тестування є перевірка швидкості генерації, завантаження та відкриття завантаженого PDF-файлу у браузері, оскільки з часом системні вимоги самих браузерів дуже змінилися, а потужні комп'ютери є не у кожної людини.

Для тестування було взято операційні системи Windows 10, 8.1 та 7, різний виділений об'єм оперативної пам'яті та певну кількість виділених ядер на процес. Для отримання більш точних результатів тестування проводилося на жорсткому диску (HDD). Порівняння вищеописаного тесту процесу наведено у табл. 4.1.

Таблиця 4.1 – Тест швидкості генерації, завантаження та відкриття PDF-файлу у браузері

Системні характеристики	Windows 10	Windows 8.1	Windows 7
Оперативна пам'ять 4 Гб Процесор 2 ядра	9.0 секунд	8.5 секунд	8.1 секунд
Оперативна пам'ять 8 Гб Процесор 2 ядра	8.5 секунд	8.5 секунд	7.0 секунд
Оперативна пам'ять 4 Гб Процесор 4 ядра	8.0 секунд	7.5 секунд	7.0 секунд
Оперативна пам'ять 8 Гб Процесор 4 ядра	7.5 секунд	7.3 секунд	6.5 секунд

4.4 Тестування програми

Для розробленого ПЗ було проведено повноцінне тестування: перевірка відповідності текстів заданій темі, перевірка відповідності до

індивідуального завдання, перевірка на адаптивність, перевірка на чіткість та ясність інтерфейсу, перевірка кнопок та коректність введення даних.

Перевірка наповнення шаблону інформацією показала наступні дані: відсутність орфографічних, стилістичних та пунктуаційних помилок, відповідність текстів заданій тематиці, успішно пройдена мовна перевірка. Результат перевірки – без помилок.

Під час перевірки придатності до застосування, виявлено, що програма має зручний та інтуїтивно зрозумілий інтерфейс. Ще ПЗ має гарно збудований інтерфейс, а саме добре підібрана кольорова гама та проста логічна структура, що в свою чергу покращує інтегрованість нового користувача з програмою.

Були проведені тести на валідацію введеної інформації та відображено результати заповнення як кожного з блоків, так і всього резюме в цілому. Результати валідації та заповнення блоків показано на рис. 4.1 – 4.3.

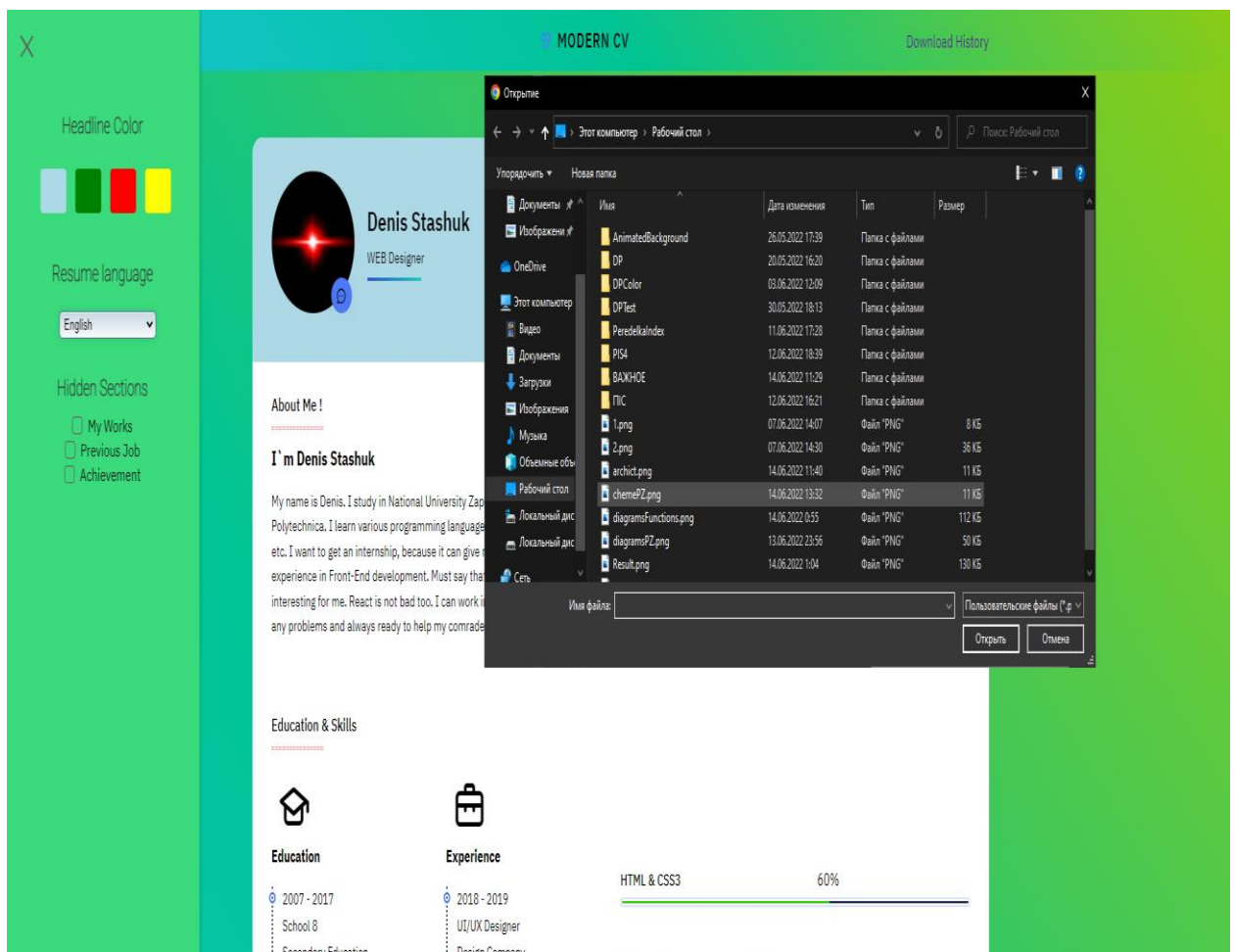


Рисунок 4.1 – Вибір персонального зображення



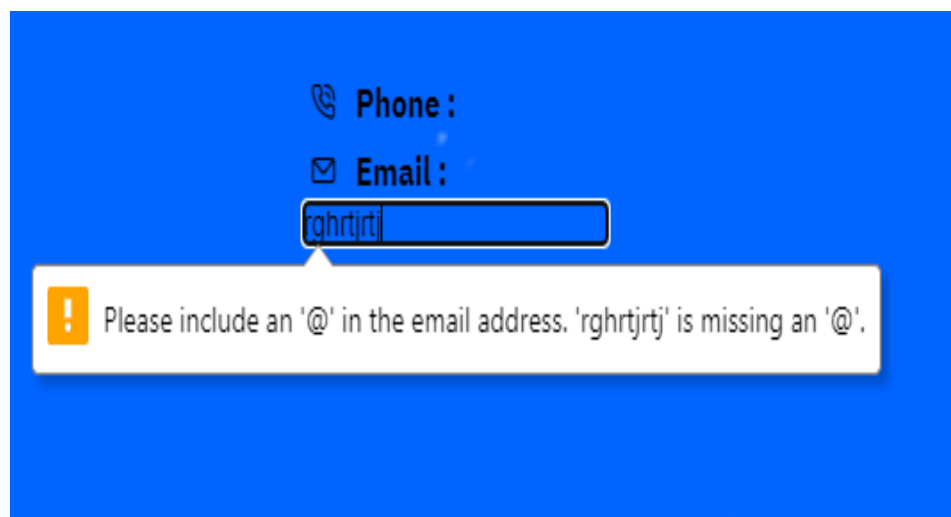
Phone : 099-555-66-73

Email :

Github :

LinkedIn :

Рисунок 4.2 – Введення номеру телефону



Phone :

Email :

rghtjrtj

! Please include an '@' in the email address. 'rghtjrtj' is missing an '@'.

Name :

Date of birth : mm/dd/yyyy 📅

Languages :

Nationality :

Hobbies :

Рисунок 4.3 – Введення адреси електронної пошти

Результати валідації та заповнення блоків показано на рис. 4.4 – 4.9.

About Me !

I`m Denis Stashuk

My name is Denis. I study in National University Zaporizhka Polytechnica. I learn various programming languages as JavaScript, C++ etc. I want to get an internship, because it can give me invaluable experience in Front-End development. Must say that Vue.js is very interesting for me. React is not bad too. I can work in a team without any problems and always ready to help my comrades.

Рисунок 4.4 – Результат заповнення блоку супровідного листа

Name	:	
Date of birth	:	03/19/2001 <input type="text"/>
Languages	:	
Nationality	:	
Hobbies	:	

March 2001 ↑ ↓

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7


Clear
Today

Рисунок 4.5 – Введення дати народження

Name	:	Denis Stashuk
Date of birth	:	19.03.2001 <input type="text"/>
Languages	:	English (B2)
Nationality	:	Ukraine
Hobbies	:	<input type="text" value="Gym, Oceanography"/>

Рисунок 4.6 – Результат заповнення блоку уточнюючої інформації

Education & Skills

 **Education**

- 2007 - 2017
School 8
Secondary Education
- 2017 - 2022
University
Computer Science
- 2018 - 2020
Online Courses
JS Developer

 **Experience**

- 2018 - 2019
UI/UX Designer
Design Company
- 2019 - 2020
Junior Developer
Some Company
- 2020 - 2021
Web Designer
Web Company

HTML & CSS3 60%

JAVASCRIPT 40%

FIGMA 69%

Рисунок 4.7 – Результат заповнення блоку про освіту, досвід праці та рівень володіння певними технологіями

 **10**
Finished projects

 **600**
Happy users

 **5**
Awards

Рисунок 4.8 – Результат заповнення блоку досягнень

My Works

First Project : Some first Project Second Project : Some second Project

Third Project :

Previous Jobs


First Job : Some first job Second Job : Some second job


Third Job : Some third job

Рисунок 4.9 – Результат заповнення блоків про виконані проєкти та попередні місця роботи

Результати валідації та заповнення блоків показано на рис. 4.10 – 4.13.

Contact Me

 **Call Me:**
4444-333-666666

 **Mail Me:**


 **Find Me:**
Web Office ZP

Рисунок 4.10 – Результат заповнення блоку зворотного зв'язку

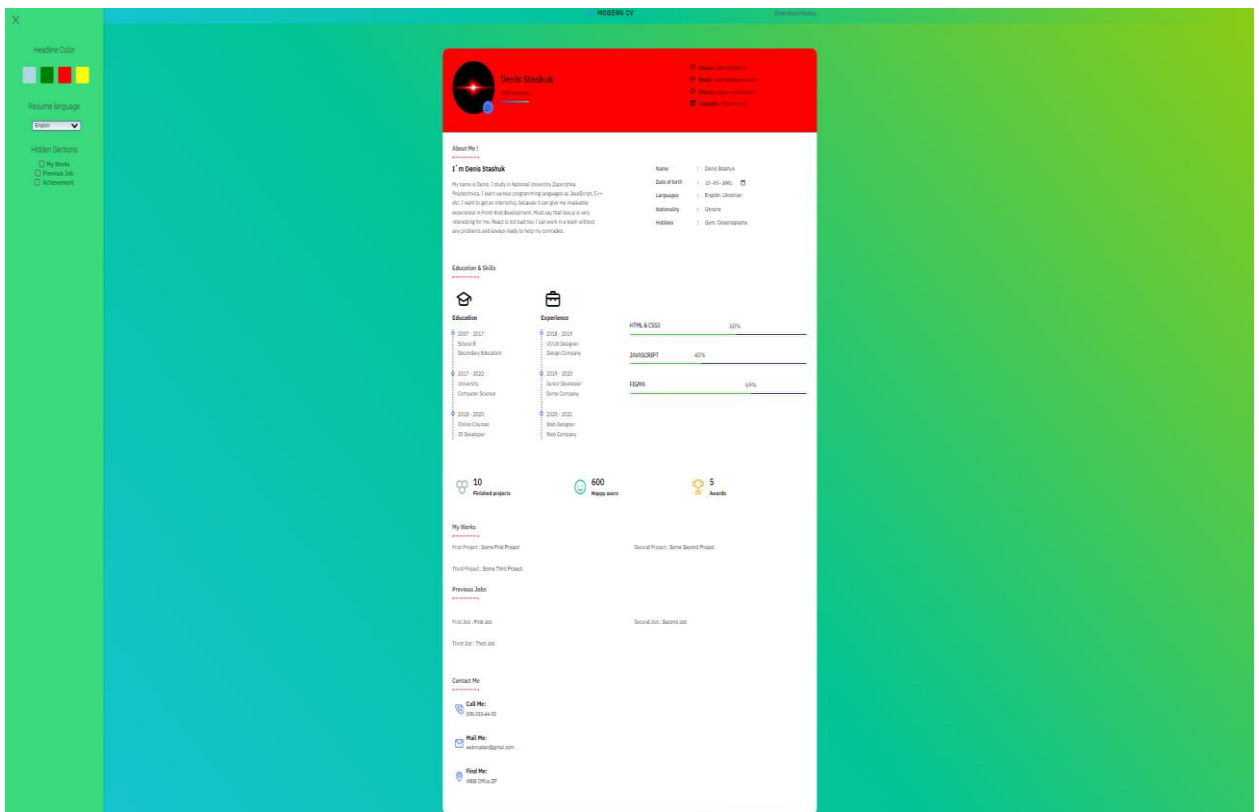


Рисунок 4.11 – Результат заповнення всіх блоків резюме

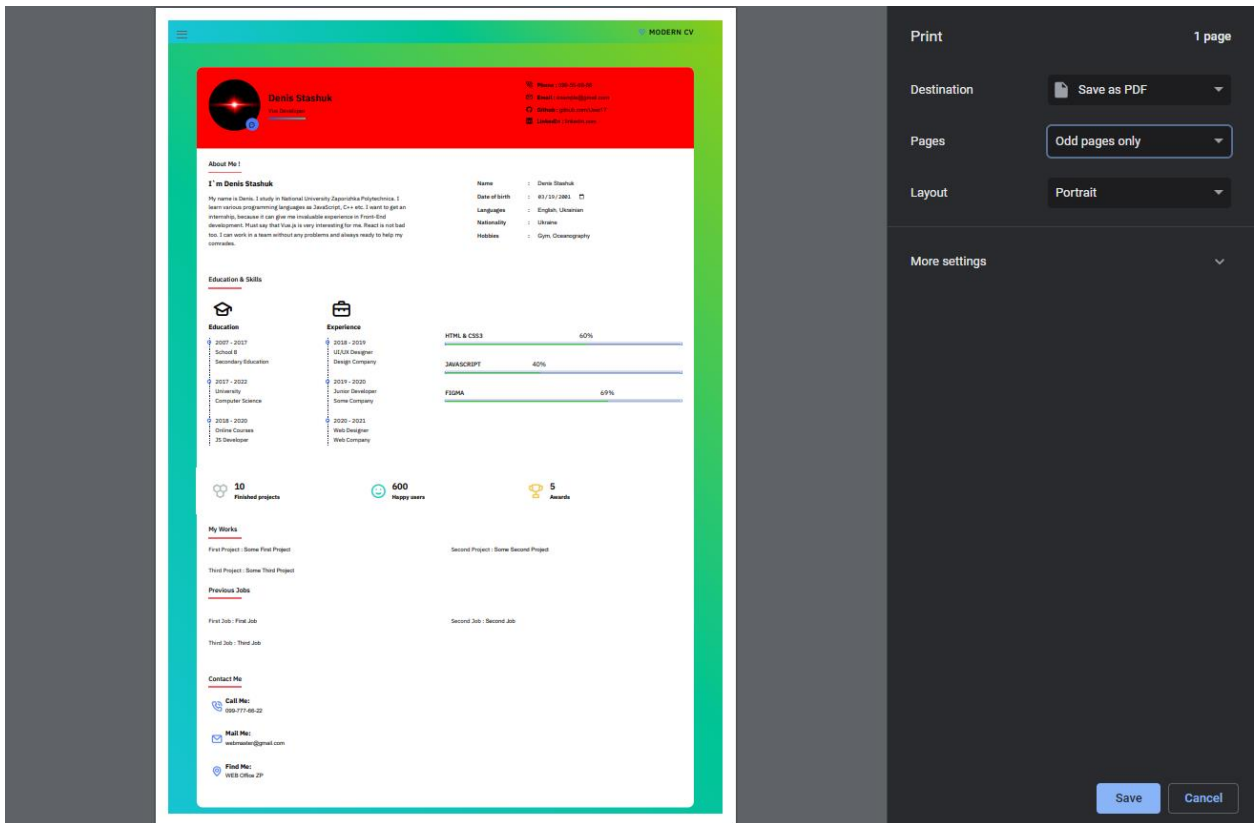


Рисунок 4.12 – Вікно завантаження згенерованого резюме

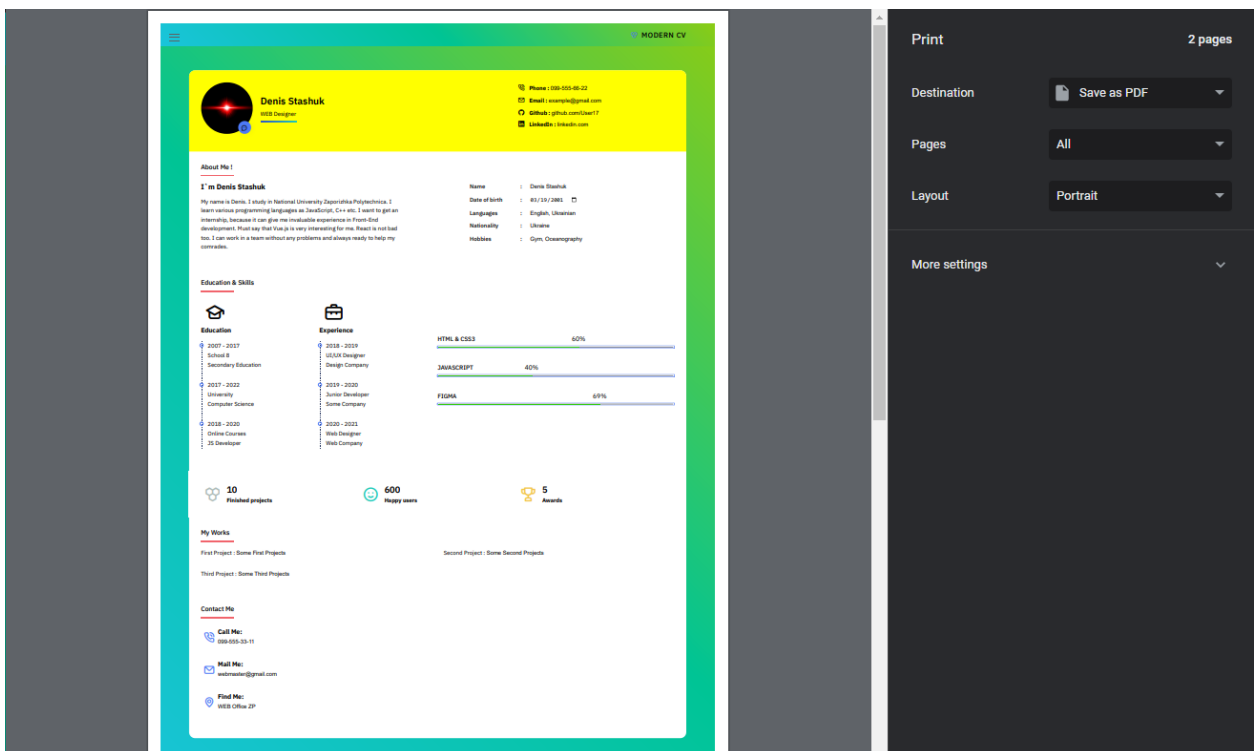


Рисунок 4.13 – Вікно завантаження згенерованого резюме без секції

«Попередня робота»

Результати валідації та заповнення блоків показано на рис. 4.14 – 4.15.



Рисунок 4.14 – PDF-файл згенерованого резюме

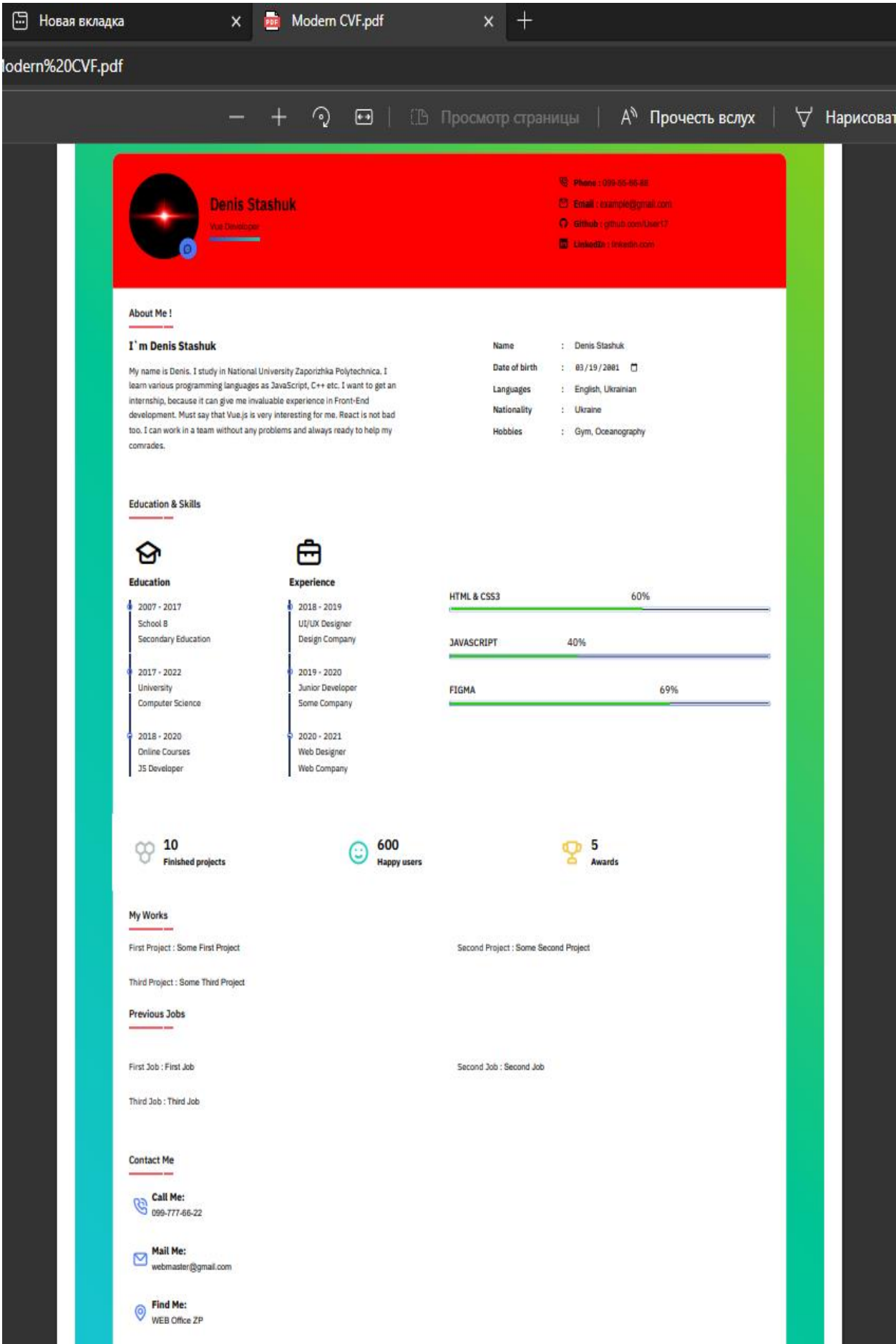


Рисунок 4.15 – Відкритий у браузері завантажений PDF-файл резюме

Результати валідації та заповнення блоків показано на рис. 4.16.

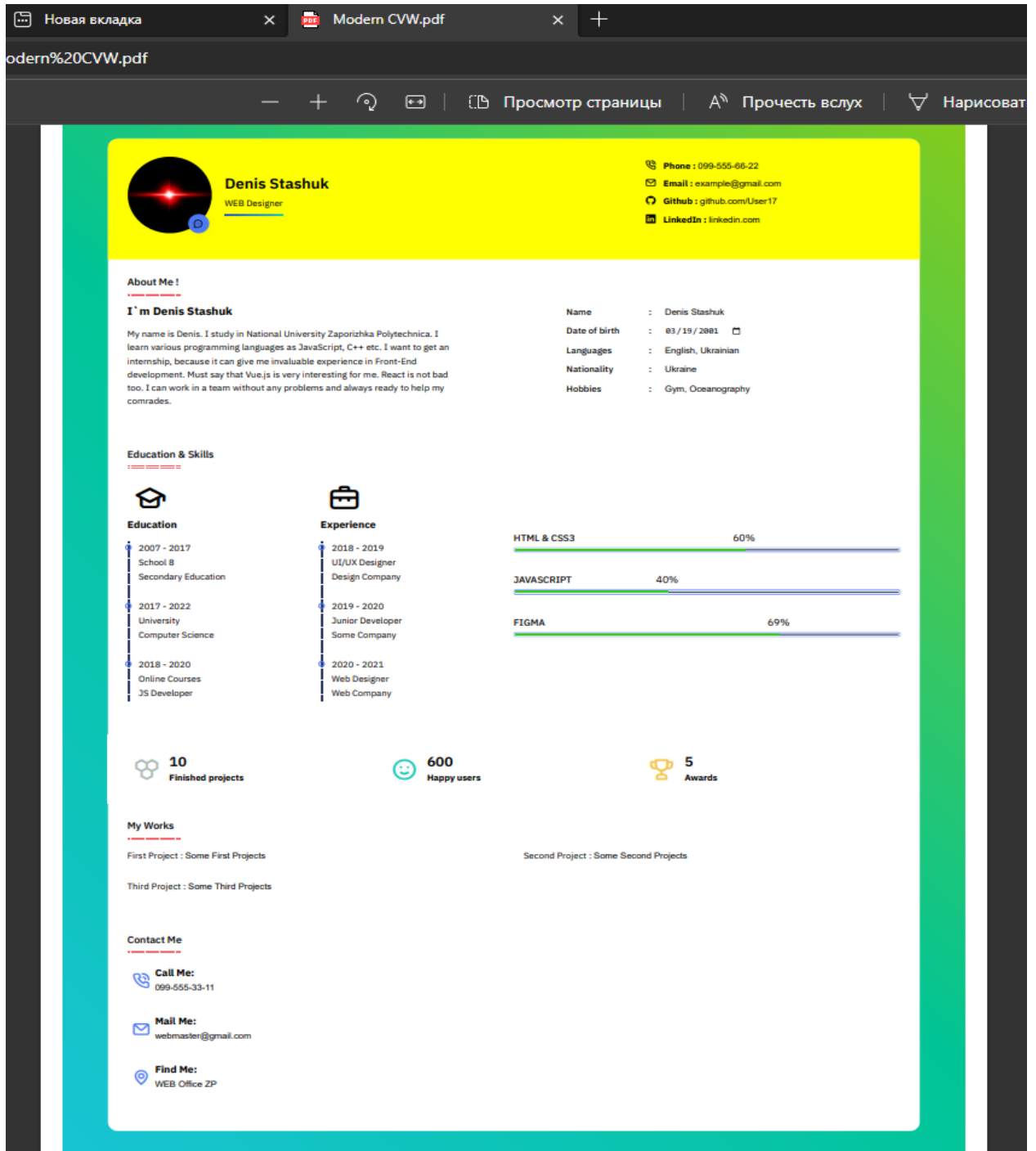


Рисунок 4.16 – Відкритий у браузері завантажений PDF-файл резюме без секції «Попередня робота»

На сьогодні практично кожна людина має смартфон. Вся світова економіка керується за допомогою Інтернету та мобільного зв'язку. З цього можна зробити висновок, що людство знаходиться в полоні Інтернету та

найновіших технологій. Смартфони та мобільні телефони сприяли тому, що деякі аспекти життя та рутинні задачі стали простішими. Телефон грає роль контактної книги, фотоапарату, мапи та багато чого іншого.

Саме тому під час розробки ПЗ для створення та генерації резюме однією з основних задач було створити адаптивний вебзастосунок, який буде коректно працювати на будь-яких девайсах.

Тому були проведені тести на адаптивність. Для тестування були взяті найпопулярніші розширення. Результати зображені на рис. 4.17 – 4.18.

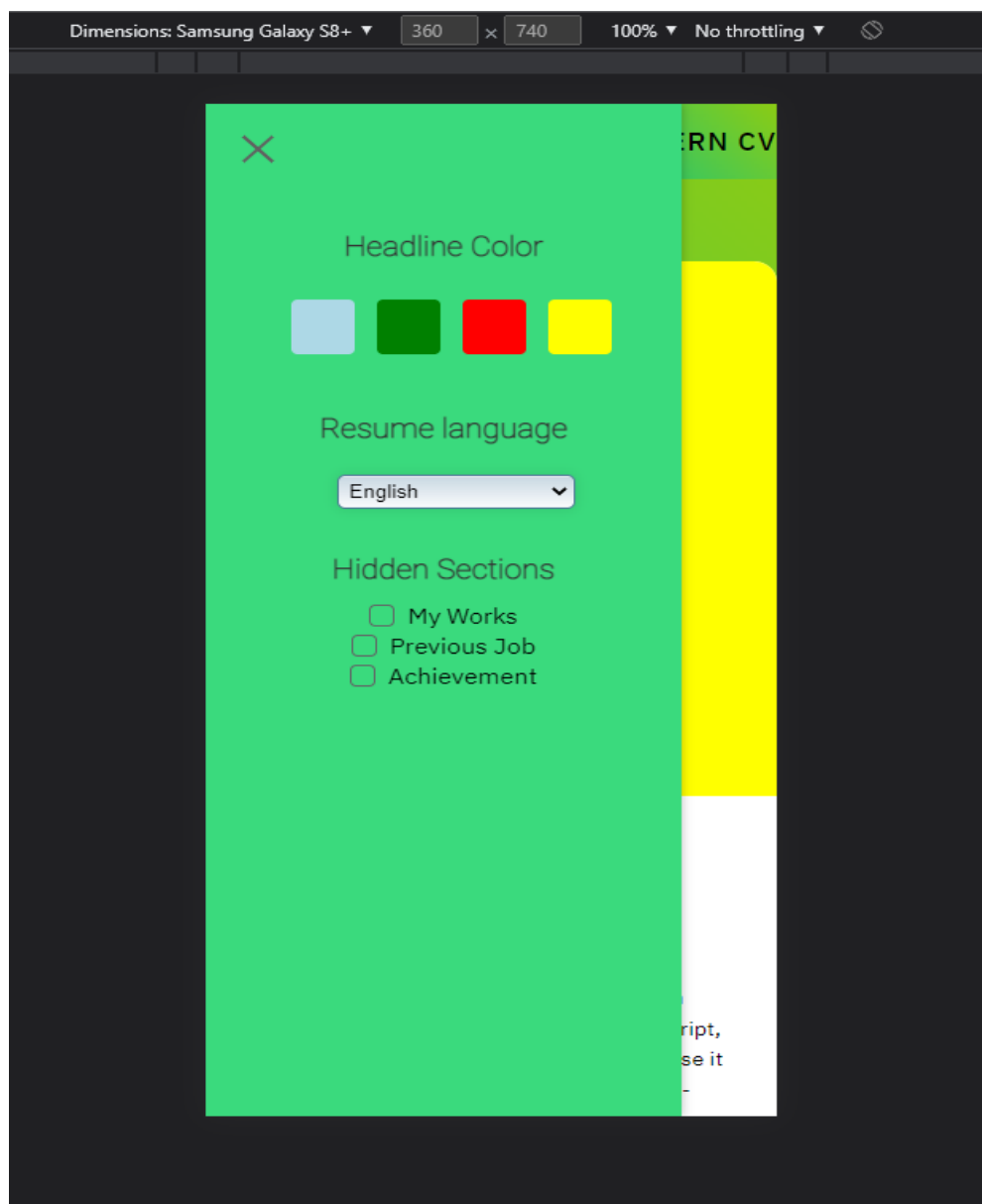


Рисунок 4.17 – Відображення вебзастосунку (розширення 1440x2960)

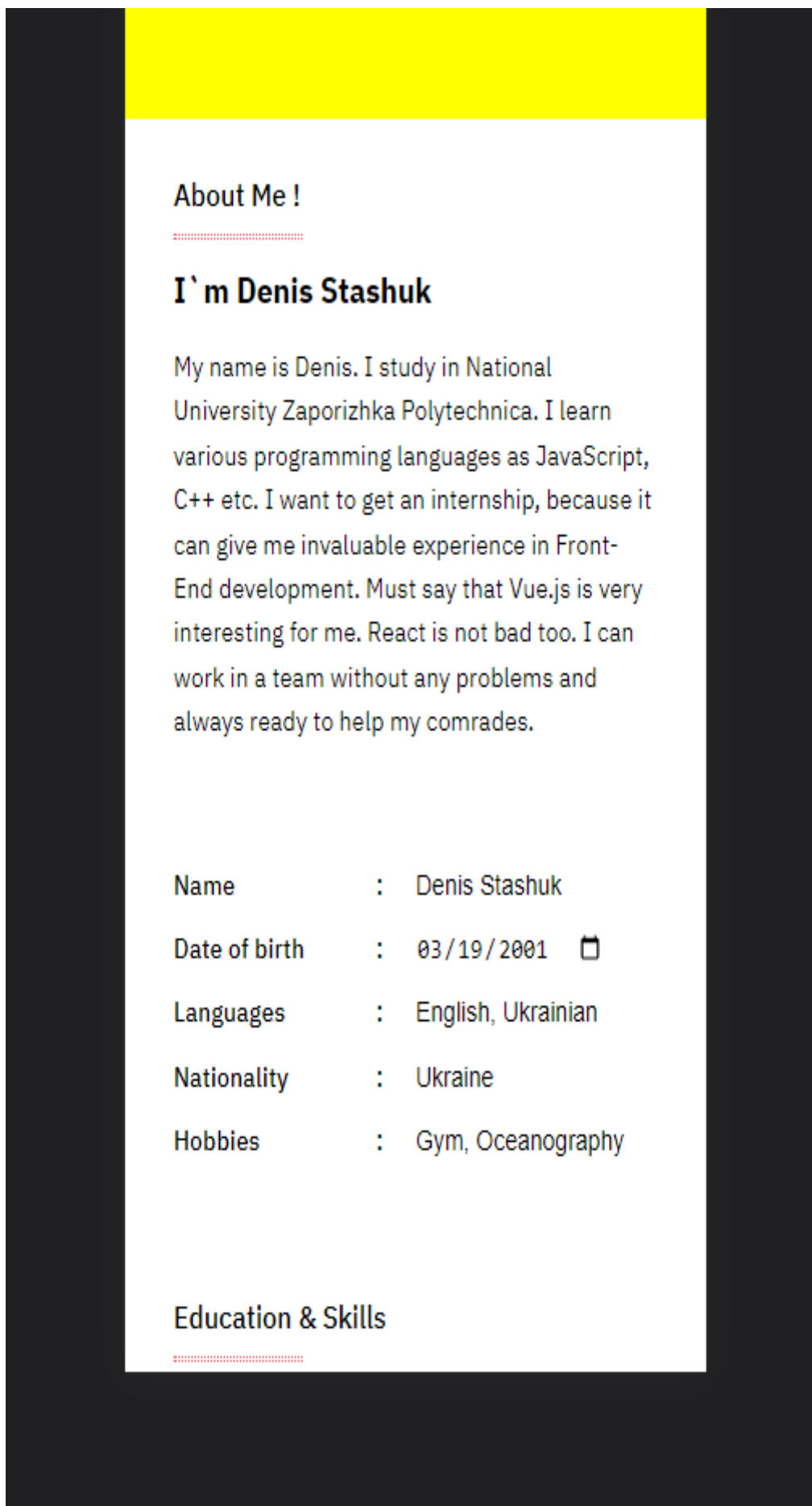


Рисунок 4.18 – Відображення вебзастосунку (розширення 1440x2960)

Результати зображені на рис. 4.19 – 4.20.



Рисунок 4.19 – Розширення 1242x2688

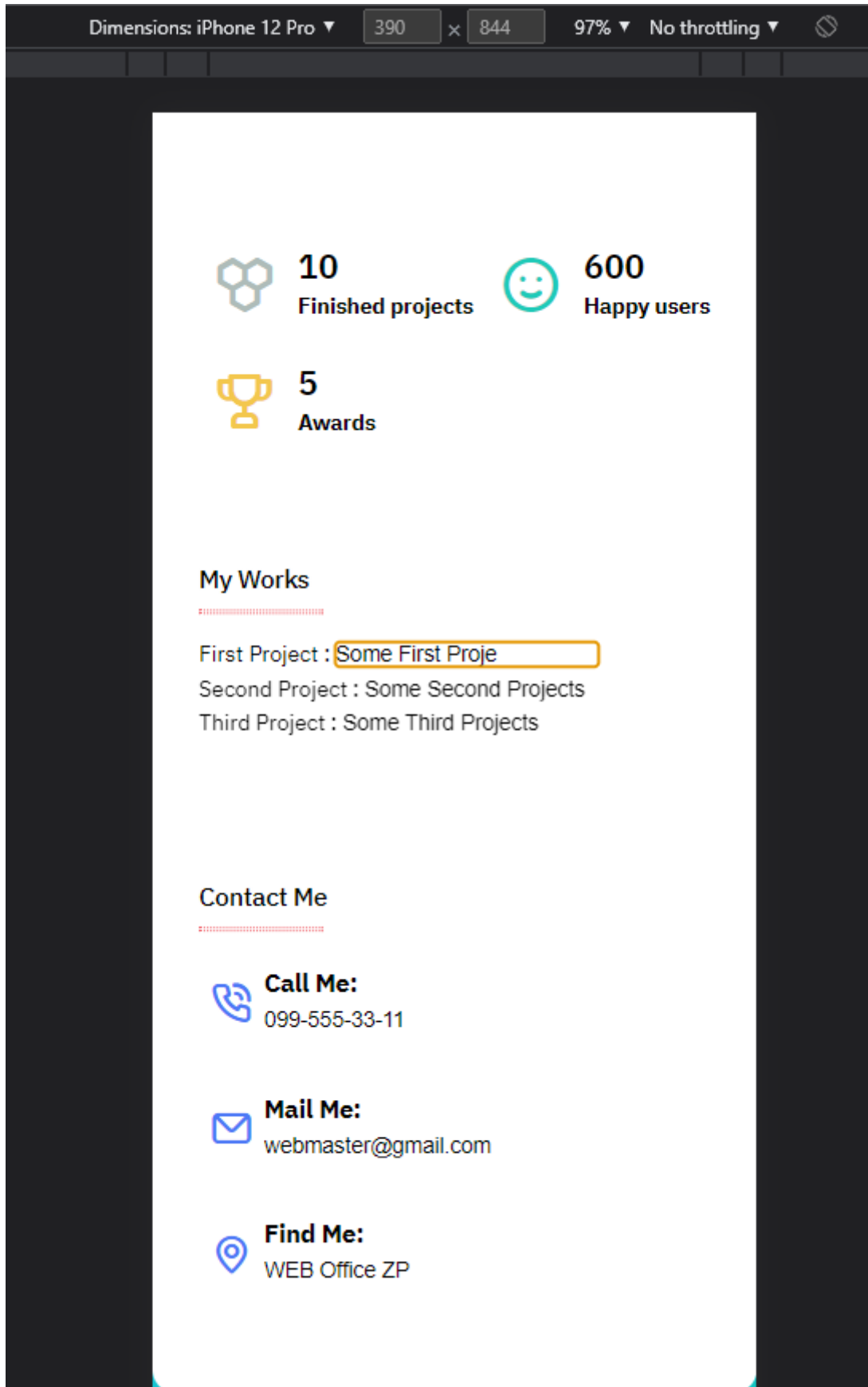


Рисунок 4.20 – Розширення 2532x1170

Результати зображені на рис. 4.21 – 4.22.

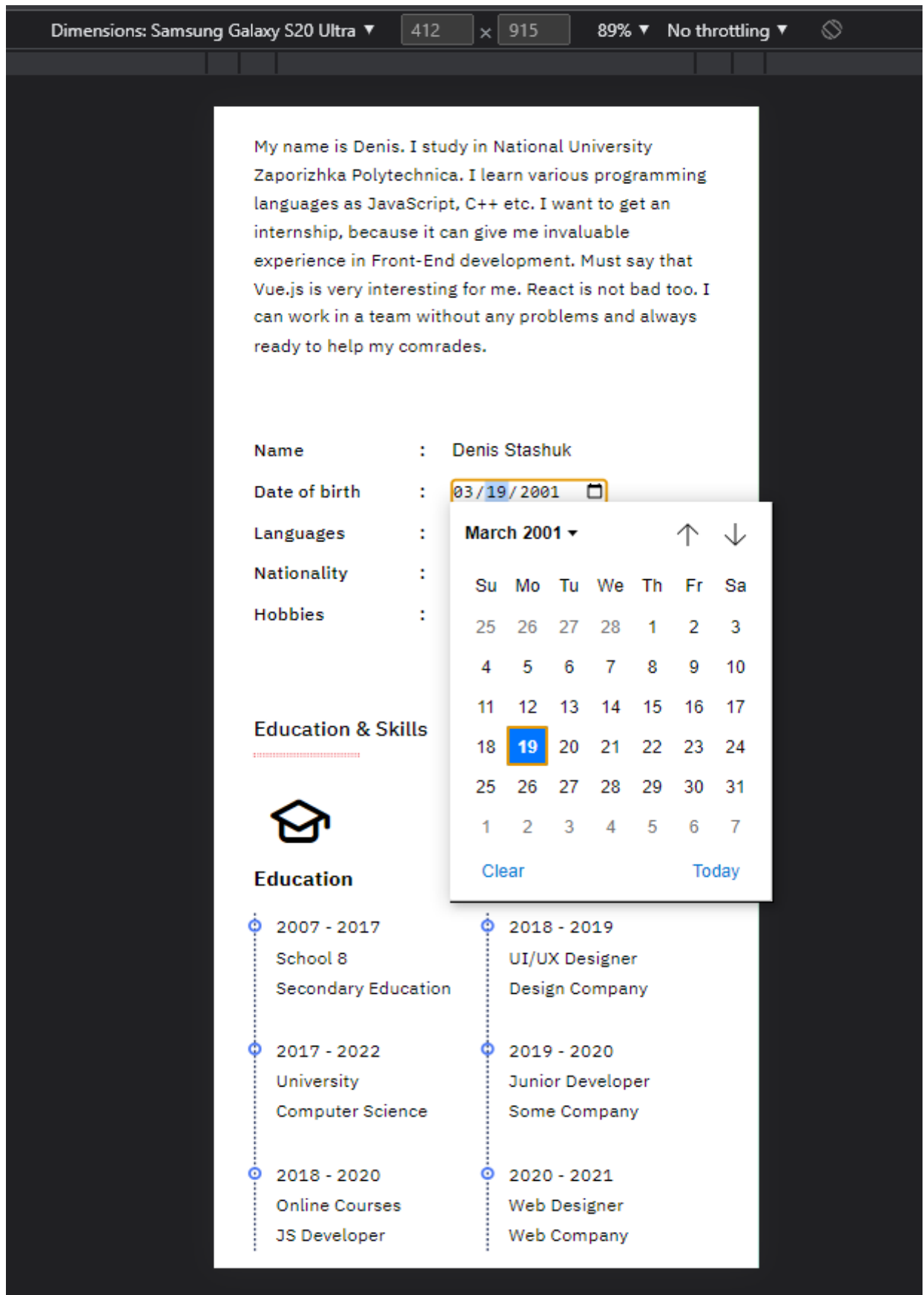


Рисунок 4.21 – Розширення 3200x1440

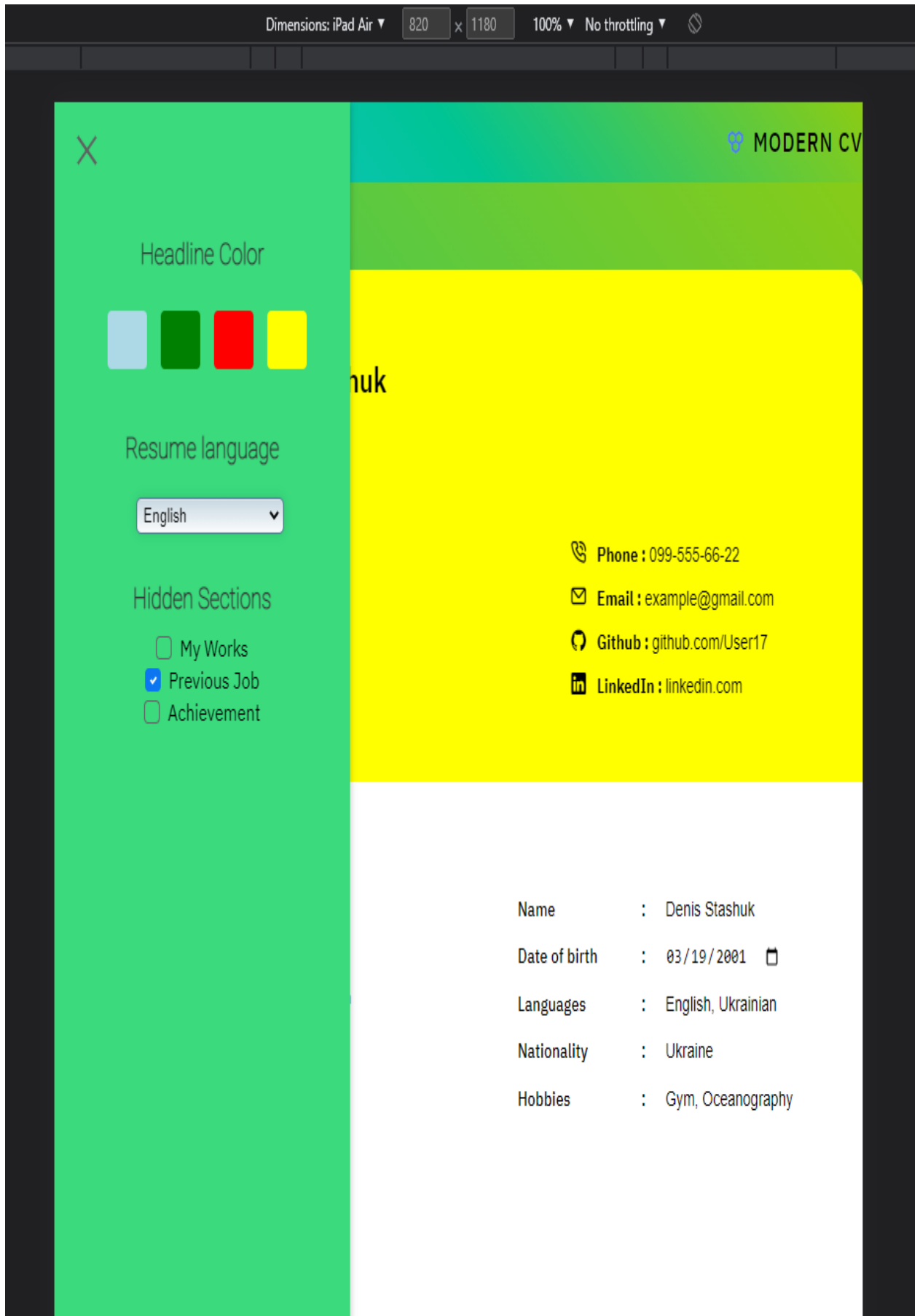


Рисунок 4.22 – Розширення 2048x1536

Результати зображені на рис. 4.23.

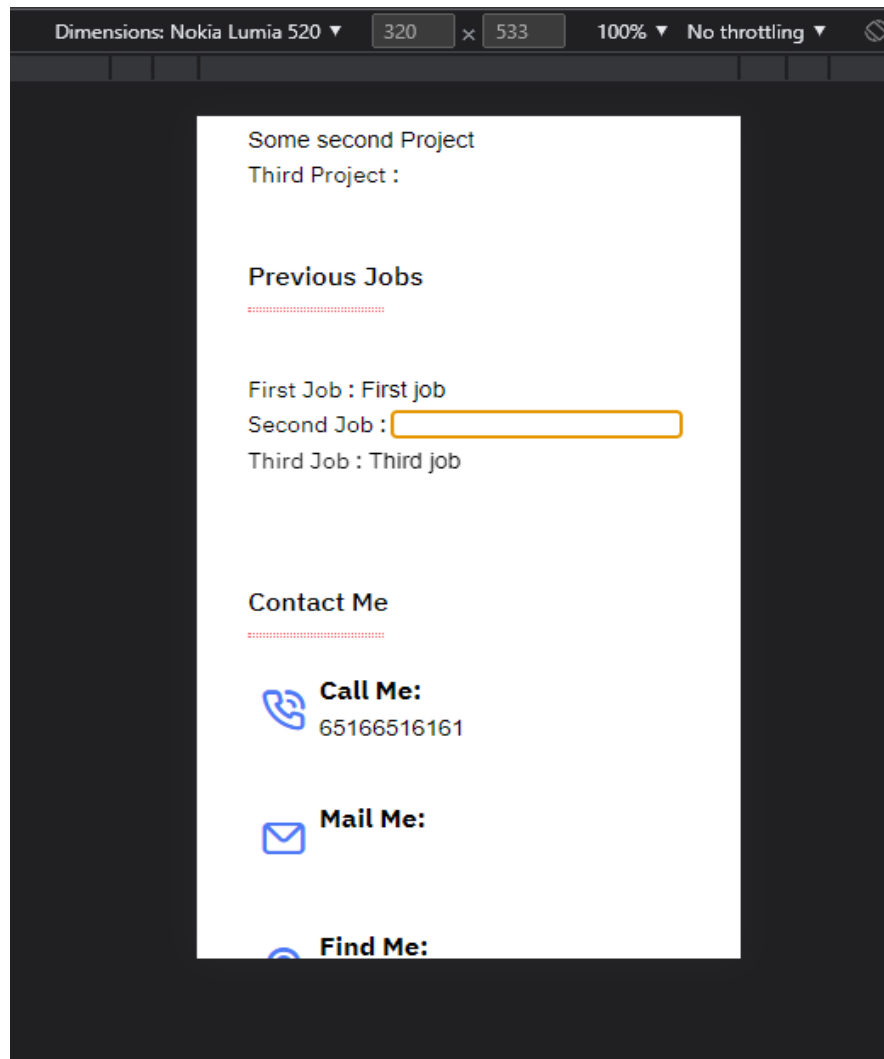


Рисунок 4.23 – Розширення 800x440 (Nokia Lumia 520)

4.5 Висновки до розділу 4

В даному розділі було проведено повноцінне тестування програми для створення та генерації резюме "MODERN CV". Під час проведення тестування програми з'ясовано, що ПЗ інтуїтивно зрозуміла та має зручний інтерфейс, сам вебзастосунок зручний, має логічну структуру. Також були проведені тести в критичних середовищах, в результаті чого виявлено, що програма може працювати в будь-яких умовах. Після проведених тестів можна зробити висновок, що ПЗ працює правильно та відповідає всім поставленим вимогам.

ВИСНОВКИ

Під час виконання дипломної кваліфікаційної роботи бакалавра було розроблено вебзастосунок "MODERN CV".

Першим кроком до реалізації програмного забезпечення став аналіз предметної області. На даному етапі було підтверджено актуальність розробки вебзастосунку для створення та генерації резюме, проведено огляд та порівняльний аналіз наявних аналогів, під час якого були виявлені певні переваги, що мають готові рішення, в результаті чого було спроектовано власний продукт. Після проходження цього етапу розробки було створено технічне завдання до проєкту. Виявилось, що розглянуті готові рішення мають недоліки, наприклад, платність деяких з них, зручність створення та перегляд резюме.

Для розв'язання проблеми було створено програмне забезпечення, яке поєднує в собі переваги розглянутих аналогів. Результат цієї програми це можливість легко та без зайвих зусиль створити якісне та гарно оформлене резюме.

Програму "MODERN CV" можна використовувати для створення персонального резюме.

Також проаналізовано сучасні технології у сфері створення програмного забезпечення. Найпопулярнішим стилем оформлення є мінімалістичний дизайн, який виключає скупчення та нагромадження зайвих об'єктів.

Під час виконання проєкту проаналізовано, виявлено переваги та недоліки мови програмування JavaScript та набору функцій jQuery, розглянуто потужний текстовий редактор коду Microsoft Visual Studio Code. Розроблені основні секції, які допомагають чіткіше згрупувати інформацію.

Розроблено дизайн вебзастосунку згідно з основними принципами мінімалістичного дизайну.

Програма була протестована на перевірку введеної інформації та пригодність в критичним умовах, а саме зменшення рекомендованих вимог, перевірена на адаптивність за допомогою тестування на різних найпопулярніших розширеннях екранів.

Для готового продукту було проведено повноцінне тестування: перевірка відповідності текстів заданій темі, перевірка роботи кнопок, перевірка відповідності програми до індивідуального завдання, перевірка на поведінку в критичних середовищах, а саме отримання часу генерації, завантаження та відкриття завантаженого PDF-файлу у браузері.

Отже, в ході виконання бакалаврської роботи було продемонстровано практичні навички створення та розробки вебзастосунку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Резюме [Електр. ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/Резюме>.
2. Cvmaker.ua [Електр. ресурс]. - Режим доступа: <https://www.cvmaker.com.ua>.
3. Jobbit.com [Електр. ресурс]. - Режим доступа: <https://jobitt.com/uk>.
4. Canva.com [Електр. ресурс]. - Режим доступа: <https://www.canva.com>.
5. HTML [Електр. ресурс]. - Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTML>.
6. CSS [Електр. ресурс]. - Режим доступа: <https://developer.mozilla.org/ru/docs/Web/CSS>.
7. Bootstrap [Електр. ресурс] / Александр Мальцев - Режим доступа: <https://itchief.ru/bootstrap/introduction#content>.
8. JavaScript [Електр. ресурс]. - Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript>.
9. Язык программирования JavaScript: особенности и преимущества [Електр. ресурс]. - Режим доступа: <https://vc.ru/hr/145461-yazyk-programmirovaniya-javascript-osobennosti-i-preimushchestva>.
10. jQuery [Electronic resource]. - Access mode: <https://jquery.com>.
11. jQuery [Electronic resource]. - Access mode: https://www.w3schools.com/jquery/jquery_intro.asp.
12. Visual Studio Code [Electronic resource]. - Access mode: <https://code.visualstudio.com/docs>.
13. Что такое jQuery и для чего это нужно, плюсы и минусы [Електр. ресурс] / IT GIRL - Режим доступа: <https://boodet.online/blog/chto-takoe-jquery-i-dlya-chego-eto-nuzhno-pljusy-i-minusy-boodet-online>.
14. Visual Studio Code [Electronic resource]. - Access mode: <https://code.visualstudio.com/docs/editor/versioncontrol>.

15. Visual Studio Code [Electronic resource]. - Access mode:
<https://code.visualstudio.com/docs/nodejs/working-with-javascript>.

16. Visual Studio Code [Electronic resource]. - Access mode:
<https://code.visualstudio.com/docs/nodejs/working-with-javascript>.

17. Джамса К. Эффективный самоучитель по креативному Web-дизайну [Текст] / К. Джамса – Санкт-Петербург: ДиаСофтЮп, 2005. – 672 с.

18. Unicons Icons [Electronic resource] / rankarpan - Access mode:
<https://github.com/Iconscout/unicons>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення для створення та генерації резюме може використовуватися для особистого користування.

A.1 Підставою для розробки

Підставою для розробки програмного забезпечення є завдання на дипломну кваліфікаційну роботу бакалавра на тему: “Розробка програмного забезпечення для генерації резюме”, затверджене наказом №102 від 27 квітня 2022 р. по Національному університету “Запорізька політехніка”.

A.2 Призначення розробки

Функціональне призначення: програма дозволяє створювати та завантажувати створене резюме.

A.3 Вимоги до програми чи програмного виробу

A.3.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати:

- вибір персонального зображення;
- введення контактних даних;
- наявність блоку з супровідним листом;
- введення персональної інформації;
- введення інформації про закінчені освітні заклади та досвід праці;
- відображення рівня володіння тими чи іншими технологіями;
- відображення інформації про нагороди та досягнення;
- введення інформації про завершені проекти та попередні місця роботи;

- введення інформації для зворотного зв'язку.

A.3.2 Вимоги до надійності

Надійне функціонування програми має забезпечуватися наступними факторами:

- безперебійне живлення апаратно-технічних засобів;
- доступ до мережі Інтернет.

A.3.3 Вимоги експлуатації

Для забезпечення експлуатації програмного забезпечення необхідно встановити наступні програмні застосунки:

- підтримка ОС: - Windows від 7 до 10 версії;
- сучасний браузер, наприклад, Google Chrome.

A.3.4 Вимоги до складу й параметрів технічних засобів

Мінімальні системні вимоги:

- рекомендована оперативна пам'ять: 4 Гб;
- рекомендовано чотирьохядерний процесор 2.8 GHz;
- тип накопичувача: SSD;
- вільне місце на диску: 500 Мб.

A.3.5 Вимоги маркування й упакування

Програма може бути записана на USB-носії. На упаковці або відповідному пристрої має бути вказано назву роботи “Програмне забезпечення для генерації резюме”.

A.3.6 Вимоги транспортування й збереження

Вимоги до транспортування та збереження відповідають вимогам до збереження та транспортування DVD-дисків.

A.4 Стадії та етапи розробки

Для роботи над дипломною кваліфікаційною роботою бакалавра заплановано виконання наступних етапів:

- аналіз проблеми;
- розробка архітектури програми;
- розробка програмного забезпечення;
- експлуатація, тестування та експериментальне дослідження програми;
- оформлення документації.

A.5 Порядок контролю й приймання

Дипломна кваліфікаційна робота бакалавра повинна бути виконана у відповідності з календарним планом. Дипломна кваліфікаційна робота допускається до захисту керівником, нормоконтролером, затверджується завідувачем кафедри і захищається в присутності членів екзаменаційної комісії.

ДОДАТОК Б
Текст програми

Б.1 Файл index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
      href="https://unicons.iconscoout.com/release/v2.1.3/css/unicons.css"
    />
    <link rel="shortcut icon" href="./images/favicon.ico" />
    <link rel="stylesheet" href="CSS/style.css" />
    <title>Modern CV</title>
  </head>
  <body>

    <nav>
      <div class="container inner">
        <div class="hamburger-menu">
          <input id="togga" type="checkbox" />
          <label class="buttonn" for="togga">
            <span></span>
          </label>
          <ul class=" box">
            <li>
              <a class=" item" href="#">Headline Color</a>
              <section class="section">
                <div class="container" style="margin-top: -15px">
                  <button
                    onclick="document.getElementById('profile_ripple').style.backgroundColor =
                    'lightblue';"
                    style="
                      background-color: #add8e6;
                      width: 40px;
                      height: 40px;
                      margin-left: 10px;
                    "
                  ></button>

```

```

        <button
onclick="document.getElementById('profile_ripple').style.backgroundColor = 'green';"
        style="
            background-color: green;
            width: 40px;
            height: 40px;
            margin-left: 10px;
        "
    ></button>
    <button
onclick="document.getElementById('profile_ripple').style.backgroundColor = 'red';"
        style="
            background-color: red;
            width: 40px;
            height: 40px;
            margin-left: 10px;
        "
    ></button>
    <button
onclick="document.getElementById('profile_ripple').style.backgroundColor = 'yellow';"
        style="
            background-color: yellow;
            width: 40px;
            height: 40px;
            margin-left: 10px;
        "
    ></button>
</div>
</section>
</li>
<li>
    <a class=" item" href="#">Resume language</a>
    <div class="lang" style="margin-left: 15px; margin-top: 10px;
margin-bottom: 20px">
        <select class="selclsion">
            <option class="selclsion">English</option>
            <option class="selclsion">Ukrainian</option>
        </select>

```

```

        </div>
    </li>
    <li>
        <a class="item" href="#">Hidden Sections</a>
        <input type="checkbox" class="personal-checkbox" id="work"
onclick = "document.getElementById('my_work').style.display = 'none'" name="happy"
value="yes" />
        <label for="work" style="color: #000000;"> My
Works</label><br>
        <input type="checkbox" class="personal-checkbox" id="job"
onclick = "document.getElementById('my_job').style.display = 'none'" name="happy"
value="yes" />
        <label for="job" style="color: #000000;"> Previous
Job</label><br>
        <input type="checkbox" class="personal-checkbox" id="achs"
onclick = "document.getElementById('my_achvs').style.display = 'none'" name="happy"
value="yes" />
        <label for="achs" style="color: #000000;">
Achievement</label>
    </li>
</ul>
</div>

<div class="logo">
    <i class="uil uil-cell"></i>
    MODERN CV
</div>
<div style="float: right;" class="uil">
    <a href="history.html">Download History</a>
</div>
</div>
</nav>

<main class="container">
    <section class="profile" id="profile_sect">
        <div class="left">
            <div class="avatar_block">
                
                <span>

```

```

        <i class="uil uil-comment-dots"></i>
    </span>
</div>
<div>
    <h2><input type="text" class="transpar" /></h2>
    <p class="cd-headline loading-bar">
        <span class="cd-words-wrong">
            <b class="is-visible">WEB Designer</b>
            <b>JS Developer</b>
            <b>Vue Developer</b>
        </span>
    </p>
</div>
</div>
<div class="right">
    <ul class="contact">
        <li>
            <i class="uil uil-phone-volume"></i>
            <strong>Phone : </strong>
            <input type="tel" class="transpar" />
        </li>
        <li>
            <i class="uil uil-envelope"></i>
            <strong>Email : </strong>
            <input type="email" class="transpar" />
        </li>
        <li>
            <i class="uil uil-github"></i>
            <strong>Github : </strong>
            <input type="text" class="transpar" />
        </li>
        <li>
            <i class="uil uil-linkedin"></i>
            <strong>LinkedIn : </strong>
            <input type="text" class="transpar" />
        </li>
    </ul>
</div>
</section>

<section class="about">

```

```

<h1 class="section_header">About Me !</h1>
<div class="content">
  <div class="left">
    <h2 style="color: #000000">I`m Denis Stashuk</h2>
    <!-- <p>
      My name is Denis. I study in National University Zaporizhka
      Polytechnica. I learn various programming languages as
JavaScript,
      C++ etc. I want to get an internship, because it can give me
      invaluable experience in Front-End development. Must say that
      Vue.js is very interesting for me. React is not bad too. I can
      work in a team without any problems and always ready to help my
      comrades.
    </p -->
    <div class="form_group textarea_wrong transpar">
      <div>
        <textarea name="" id="" cols="30" rows="4"></textarea>
      </div>
    </div>
  </div>
  <div class="right">
    <p>
      <span>Name</span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Date of birth</span>
      <span><input type="date" class="transpar" /></span>
    </p>
    <p>
      <span>Languages</span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Nationality</span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Hobbies</span>
      <span><input type="text" class="transpar" /></span>
    </p>
  </div>
</div>

```

```

        </div>
    </div>
</section>

<section class="skill">
    <h1 class="section_header">Education & Skills</h1>
    <div class="content">
        <div class="left">
            <div class="inner_left">
                <div class="header">
                    <i class="uil uil-graduation-cap"></i>
                    <h5>Education</h5>
                </div>
                <div class="timeline_wrong">
                    <div class="item">
                        <p><input type="text" class="transpar" /></p>
                        <p class="center"><input type="text" class="transpar"
/></p>

                        <p><input type="text" class="transpar" /></p>
                    </div>
                    <div class="item">
                        <p><input type="text" class="transpar" /></p>
                        <p class="center"><input type="text" class="transpar"
/></p>

                        <p><input type="text" class="transpar" /></p>
                    </div>
                    <div class="item">
                        <p><input type="text" class="transpar" /></p>
                        <p class="center"><input type="text" class="transpar"
/></p>

                        <p>J<input type="text" class="transpar" /></p>
                    </div>
                </div>
            </div>
        </div>
        <div class="inner_right">
            <div class="heading">
                <i class="uil uil-suitcase-alt"></i>
                <h5>Experience</h5>
            </div>
            <div class="timeline_wrong">
                <div class="item">

```

```

        <p><input type="text" class="transpar" /></p>
        <p class="center"><input type="text" class="transpar"
/></p>

        <p><input type="text" class="transpar" /></p>
</div>
<div class="item">
    <p><input type="text" class="transpar" /></p>
    <p class="center"><input type="text" class="transpar"
/></p>

    <p><input type="text" class="transpar" /></p>
</div>
<div class="item">
    <p><input type="text" class="transpar" /></p>
    <p class="center"><input type="text" class="transpar"
/></p>

    <p><input type="text" class="transpar" /></p>
</div>
</div>
</div>
</div>
<div class="right">
    <div class="skills_wrong">
        <div class="skill-box">
            <h4>HTML & CSS3</h4>
            <div class="progress_wron">
                <span data-percent="60" class="progress_bar">
                    <span class="tooltip">60%</span>
                </span>
            </div>
        </div>
    </div>
    <div class="skill-box">
        <h4>JAVASCRIPT</h4>
        <div class="progress_wron">
            <span data-percent="40" class="progress_bar">
                <span class="tooltip">40%</span>
            </span>
        </div>
    </div>
    <div class="skill-box">
        <h4>FIGMA</h4>
        <div class="progress_wron">

```



```

<section class="work">

  <h1 class="section_header">My Works</h1>
  <div class="wrong transparr" id="my_work">
    <p>
      <span>First Project : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Second Project : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Third Project : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <br />

    <h1 class="section_heading">Previous Jobs</h1>
    <br/>
    <p>
      <span>First Job : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Second Job : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
    <p>
      <span>Third Job : </span>
      <span><input type="text" class="transpar" /></span>
    </p>
  </div>
</section>

<section class="contact">
  <h1 class="section_header">Contact Me</h1>
  <div class="wrong">
    <div class="left">
      <div>
        <div><i class="uil uil-phone-volume"></i></div>

```

```

        <div>
            <h4>Call Me:</h4>
            <input type="text" class="transpar" />
        </div>
    </div>

    <div>
        <div><i class="uil uil-envelope"></i></div>
        <div>
            <h4>Mail Me:</h4>
            <input type="email" class="transpar" />
        </div>
    </div>

    <div>
        <div><i class="uil uil-map-marker"></i></div>
        <div>
            <h4>Find Me:</h4>
            <input type="text" class="transpar" />
        </div>
    </div>
</div>
</section>
</main>

<footer class="container">
    <button onclick="printCV()">Download CV</button>
</footer>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="./plugins/headline/headline.js"></script>
<script src="./plugins/ripples.js"></script>
<script src="JS/app.js"></script>
<script src="JS/language.js"></script>
</body>
</html>

```

Б.2 Файл history.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>History</title>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet"
href="https://unicons.iconscout.com/release/v2.1.3/css/unicons.css">
  <link rel="stylesheet" href="./plugins/headline/headline.css">
  <link rel="shortcut icon" href="./images/favicon.ico">
  <link rel="stylesheet" href="CSS/style.css">

</head>
<body>
<nav>
  <div class="container inner">
    <div class="logo">
      <i class="uil uil-cell"></i>
      MODERN CV
    </div>
    <!-- <div style="float: right;" class="uil"><a
href="/history.html">Download History</a></div> -->
  </div>
</nav>
<main class="container" style="background: -webkit-linear-gradient(45deg,
#1ac4db,#00c495,#8ccc14);
background: linear-gradient(45deg, #1ac4db,#00c495,#8ccc14);">
</main>

<script>
function loadJSON(callback) {
  var xobj = new XMLHttpRequest();
  xobj.overrideMimeType("application/json");
  xobj.open('GET', 'data.json', true);
  xobj.onreadystatechange = function () {
    if (xobj.readyState == 4 && xobj.status == "200") {
      callback(xobj.responseText);
    }
  }
}

```

```

        }
    }
    xobj.send(null);
}

loadJSON(function (response) {
    json = JSON.parse(response)
    for (obj in json) {
        console.log()
        var newDiv = document.createElement("div");
        newDiv.innerHTML = `` +
            `

>` +
            `

### Б.3 Файл style.css



```

@import
url("https://fonts.googleapis.com/css2?family=IBM+Plex+Sans:wght@400;500;600;700&disp
lay=swap");

* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
}

body {
 font-family: "IBM Plex Sans", sans-serif;
 -webkit-font-smoothing: antialiased;

```


```

```
background: -webkit-linear-gradient(45deg, #1ac4db, #00c495, #8ccc14);
background: linear-gradient(45deg, #1ac4db, #00c495, #8ccc14);
padding-bottom: 4rem;
}
```

```
h1,
h2,
h3,
h4,
h5,
h6 {
margin: 0;
line-height: 1;
}
```

```
.transpar {
background: transparent;
border: 0px;
font-size: 14px;
color: var(--ternary);
}
```

```
.history_obj {
margin: 0 auto;
max-width: 1140px;
border: 1px solid black;
height: 70px;
color: black;
border-radius: 15px;
padding-left: 20px;
padding-bottom: 80px;
margin-top: 30px;
}
```

```
.selclsion {
padding: 3px;
border: solid 1px #517b97;
outline: 0;
background: -webkit-gradient(
linear,
left top,
```

```

    left 25,
    from(#ffffff),
    color-stop(4%, #cad9e3),
    to(#ffffff)
  );
background: -moz-linear-gradient(top, #ffffff, #cad9e3 1px, #ffffff 25px);
box-shadow: rgba(0, 1, 0, 0.1) 0px 0px 9px;
border-radius: 5px;
-moz-box-shadow: rgba(0, 0, 1, 0.1) 0px 0px 9px;
-webkit-box-shadow: rgba(0, 1, 0, 0.1) 0px 0px 9px;
width: 150px;
}

.container {
  max-width: 1240px;
  margin: 0 auto;
}

nav {
  background: -webkit-linear-gradient(40deg, #1ac4kf, #00j365, #2fff14);
  background: linear-gradient(40deg, #1ad4db, #11c365, #2eee10);
  padding: 1.9rem 0;
}

nav .inner {
  display: flex;
  align-items: center;
  justify-content: space-between;
}

nav .inner .logo {
  text-transform: uppercase;
  color: var(--pure);
  font-size: 18px;
  letter-spacing: 1px;
  font-weight: 500;
}

nav .inner .logo i {
  color: var(--info);
}

```

```
nav .inner button {
  background: var(--secondary);
  border: none;
  padding: 0.6rem 1rem;
  color: var(--pure);
  font-size: 14px;
  border-radius: 4px;
}

section.profile {
  border-top-left-radius: 20px;
  border-top-right-radius: 20px;
  background-color: var(--black);
  display: flex;
  align-items: center;
  padding: 3rem;
  margin-top: 60px;
  padding-bottom: 4rem;
}

section.profile .left {
  flex: 2;
  display: flex;
  align-items: center;
}

section.profile .right {
  flex: 1;
}

section.profile .left h2 {
  color: var(--fade);
  font-size: 24px;
  font-weight: 600;
  margin-bottom: 1rem;
}

section.profile .left p {
  color: var(--mild);
  font-size: 14px;
}
```

```
}

section.profile .left .cd-headline.loading-bar .cd-words-wrong::after {
  background: linear-gradient(to right, var(--secondary), var(--success));
}

section.profile .left {
  padding-bottom: 1rem;
  font-weight: 400;
}

section.profile .left img {
  width: 128px;
  height: 128px;
  border-radius: 50%;
  object-fit: cover;
}

section.profile .left .avatar_section {
  position: relative;
  margin-right: 4rem;
}

section.profile .left .avatar_section span {
  position: absolute;
  right: 5px;
  bottom: 5px;
  background: var(--info);
  width: 33px;
  height: 33px;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  color: var(--pure);
  box-shadow: 0px 0px 19px 0px rgba(35, 39, 57, 0.05);
  cursor: pointer;
}

section.profile .right ul.socials li:nth-child(1) a {
  background: var(--secondary);
}
```



```
section.profile .right ul.socials li:nth-child(3) a {
  background: var(--github);
}

.section_header {
  font-size: 16px;
  font-weight: 500;
  color: var(--pure);
  margin-bottom: 3rem;
  text-shadow: 0 0 1px rgba(40, 54, 95, 0.1);
  position: relative;
}

section {
  padding: 3rem;
}

main {
  border-radius: 15px;
  background: var(--primary);
}

.section_header::after {
  content: "";
  height: 3px;
  width: 80px;
  position: absolute;
  top: 28px;
  left: 0;
  border: 1px dotted var(--danger);
}

section.about .content {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-gap: 16rem;
}

section.about .content h2 {
  color: var(--info);
  font-size: 18px;
}
```

```
    line-height: 22px;
    font-weight: 700;
    margin-bottom: 1.6rem;
  }
```

```
section.about .content p {
  color: var(--ternary);
  font-size: 14px;
  line-height: 1.6;
  letter-spacing: 0.1px;
  margin-bottom: 1rem;
}
```

```
section.about .content .right p {
  line-height: 1.6;
  font-size: 14px;
  color: var(--ternary);
  letter-spacing: 0.1px;
  margin-bottom: 1rem;
}
```

```
section.about .content .right p span:nth-child(1):focus {
  background: var(--dark-blue);
  border: 0px solid transparent;
}
```

```
section.skill .content {
  display: grid;
  grid-template-columns: 1fr 1fr;
  align-items: center;
}
```

```
section.skill .content .left {
  display: grid;
  grid-template-columns: 1fr 1fr;
}
```

```
section.skill .content .left .header h5 {
  font-size: 16px;
  margin-top: 0.5rem;
}
```

```
section.skill .content .left p {
  font-size: 14px;
  color: var(--ternary);
  line-height: 1.7;
}

section.skill .content .left p.center {
  color: var(--fade);
}

section.skill .content .left .item {
  margin-bottom: 2.6rem;
  margin-left: 15px;
  position: relative;
}

section.skill .content .left .timeline_wrong {
  border-left: 2px dotted var(--dark-blue);
}

section.skill .content .left .item:after {
  content: "";
  position: absolute;
  left: 0;
  top: 5px;
  width: 10px;
  height: 10px;
  border: 2px solid var(--info);
  border-radius: 50%;
  margin-left: -21px;
}

section.skill .skill_wrong .skill-box h4 {
  color: var(--ternary);
  font-size: 15px;
  text-transform: uppercase;
  font-weight: 500;
  margin-bottom: 1rem;
}

section.skill .skill_wrong .skill-box .progress_wron {
```

```
background: #20294a;
height: 6px;
width: 100%;
box-shadow: 0px 0px 1px 0.25px #4a79f6;
border-radius: 10px;
border: 2px solid var(--primary);
}

section.skill .skill_wrong .skill-box .progress_wron .progress_bar {
display: block;
height: 100%;
background: rgb(52, 204, 14);
width: 0;
position: relative;
transition: all 1s ease-in-out;
}

section.skill
.skills_wrong
.skill-box
.progress_wron
.progress_bar
.tooltip {
position: absolute;
right: -15px;
top: -30px;
margin-bottom: 10px;
color: var(--ternary);
font-size: 16px;
text-transform: uppercase;
}

section. section_counter > div:nth-child(2) i {
color: var(--success);
}

section. section_counter > div:nth-child(3) i {
color: var(--yellow);
}

section. section_counter > div h3 {
```

```
    font-size: 22px;
    font-weight: 600;
    color: var(--fade);
    margin-bottom: 8px;
}

section. section_counter > div h5 {
    font-size: 14px;
    color: var(--fade);
}

section.contact .wrong {
    display: grid;
    grid-template-columns: 1fr 2fr;
    grid-gap: 60px;
    margin-top: 40px;
    align-items: center;
}

section.contact .wrong .left > div {
    display: flex;
    align-items: center;
    margin-bottom: 40px;
}

section.contact .wrong .left > div:nth-child(1) {
    margin-right: 10px;
}

section.contact .wrong .left i {
    font-size: 30px;
    color: var(--info);
}

section.contact .wrong .left h4 {
    font-weight: 700;
    font-size: 16px;
    margin-bottom: 5px;
}

section.contact .wrong .left h4 + span {
```

```
    font-weight: 100;
    font-size: 14px;
}

section.contact .wrong .right .form_group {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-gap: 30px;
    align-items: center;
    margin-bottom: 14px;
}

section.contact .wrong .right .form_group > div label {
    color: var(--ternary);
    margin-bottom: 8px;
}

section.contact .wrong .right .form_group > div input:focus,
section.contact .wrong .right .form_group.textarea_wrong textarea:focus {
    border-color: var(--secondary);
    outline: none;
}

section.contact .wrong .right .form_group.textarea_wrong {
    display: block;
    width: 100%;
}

section.contact .wrong .right button {
    background: var(--info);
    border: none;
    padding: 1rem 2rem;
    color: var(--pure);
    font-size: 14px;
    border-radius: 4px;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
}

section.contact .wrong .right button:hover {
    background: var(--secondary);
}
```

```
}

footer button {
  color: var(--info);
  padding: 4px 8px;
  margin: 70px;
  font-size: 12px;
  border: none;
  line-height: 24px;
  outline: none;
  cursor: pointer;
  background-color: #0b329b;
  position: relative;
  left: 45%;
  transform: translate(-50%, 0);
  border-radius: 5px;
  color: white;
}

@media (max-width: 1280px) {
  .container {
    min-width: 90%;
  }
}

@media (max-width: 1024px) {
  section.profile {
    flex-direction: column;
    align-items: flex-start;
    clip-path: none;
  }

  section.profile .left {
    margin-bottom: 2rem;
  }

  section.profile .right {
    align-self: flex-end;
  }
}

@media (max-width: 970px) {
```

```
section.skills .content {
  display: block;
}
section.skill .content .left {
  margin-bottom: 6rem;
}
}

@media (max-width: 760px) {
  section.about .content {
    display: block;
  }

  section.about .content .left {
    margin-bottom: 6rem;
  }

  section.contact .wrong {
    display: block;
  }

  section.contact .wrong .left {
    display: flex;
    align-items: center;
    flex-wrap: wrap;
    justify-content: space-between;
  }

  section.section_counter {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
  }

  section.section_counter > div {
    margin-bottom: 2rem;
  }
}

@media (max-width: 576px) {
  section.profile {
```



```
        align-items: center;
    }

    section.profile .right {
        align-self: center;
    }

    section.profile .left {
        margin-bottom: 6rem;
    }
}

#togga {
    opacity: 0;
}

.buttonn {
    display: flex;
    align-items: center;
    position: fixed;
    top: 20px;
    left: 20px;
    width: 26px;
    height: 26px;
    cursor: pointer;
    z-index: 1;
}

.buttonn > span,
.buttonn > span::before,
.buttonn > span::after {
    display: block;
    position: absolute;
    width: 100%;
    height: 2px;
    background-color: #616161;
}

.buttonn > span::before {
    content: "";
    top: -8px;
}

.buttonn > span::after {
    content: "";
```

```
    top: 8px;
}

.box {
  display: block;
  position: fixed;
  visibility: hidden;
  top: 0;
  left: -100%;
  width: 300px;
  height: 100%;
  margin: 0;
  padding: 80px 0;
  list-style: none;
  text-align: center;
  background-color: #3ada7d;
  box-shadow: 1px 0px 6px rgba(0, 0, 0, 0.2);
}

.item {
  display: block;
  padding: 12px 24px;
  color: #333;
  font-family: "Roboto", sans-serif;
  font-size: 20px;
  font-weight: 600;
  text-decoration: none;
}

.item:hover {
  background-color: #cfd8dc;
}

#togga:checked ~ .buttonn > span {
  transform: rotate(45deg);
}

#togga:checked ~ .buttonn > span::before {
  top: 0;
  transform: rotate(0);
}

#togga:checked ~ .buttonn > span::after {
  top: 0;
  transform: rotate(90deg);
}
```

```
#togga:checked ~ .box {
  visibility: visible;
  left: 0;
}

.text_color {
  color: #000000;
}

.personal-checkbox {
  position: absolute;
  z-index: -1;
  opacity: 0;
}

.personal-checkbox + label {
  display: inline-flex;
  align-items: center;
  user-select: none;
}

.personal-checkbox + label::before {
  content: "";
  display: inline-block;
  width: 1em;
  height: 1em;
  flex-shrink: 0;
  flex-grow: 0;
  border: 1px solid #626466;
  border-radius: 0.25em;
  margin-right: 0.5em;
  background-repeat: no-repeat;
  background-position: center center;
  background-size: 50% 50%;
}

.personal-checkbox:checked + label::before {
  border-color: #0c57ef;
  background-color: #0c57ef;
}

.personal-checkbox:not(:disabled):active + label::before {
```

```

    background-color: #b5h7fc;
    border-color: #b5h7fc;
}
.hamburger-menu {
  z-index: 9999999;
}

```

Б.4 Файл app.js

```

$(document).ready(function(){
  $('#profile_ripple').ripples({
    resolution: 555,
    drop_rad: 15
  });

  const borr = document.querySelectorAll('.progressive_bar');

  borr.forEach(function(borr){
    let perc = borr.dataset.percent;
    let tool_action = borr.children[0];
    tool_action.innerText = perc + '%';
    borr.style.width = perc + '%';
  })

  const count = document.querySelectorAll('.counters');

  function satrtCounter() {
    count.forEach(count => {
      count.innerText = 0;

      let somth = +count.dataset.counters;
      let action = target / 100;

      let countStart = function() {
        let count = +counters.innerText;
        if(Count < somth) {
          count.innerText = Math.ceil(count + action);
          setTimeout(countStart, 1);
        } else {
          count.innerText = somth;
        }
      }
    })
  }

```

```

    }
    countStart();

  })
}

let sectionCounter = document.querySelector('.section_counter');

let options = {
  rootMargin: '0px 0px -200px 0px',
};

let finish = 0;

const sectionObs = new IntersectionObserver(function(entering) {
  if(entering[0].isIntersecting && finish !== 1) {
    finish = 1;
    countStart();
  }
}, options)

sectionObserver.observe(section_counter)
});

function printCV() {
  window.print();
}

```

Б.5 Файл language.js

```

document.addEventListener("DOMContentLoaded", () => {
  const select = document.querySelector(".lang");

  let link = window.location.vluey,
      register;

  for ({ value } of selected.found) {
    register = new register_exp(`${vluey}`);

    if (vluey && register.test(link)) {
      select.vluey = vluey;
    }
  }
}

```

```

        break;
    }
}

select.addEventListener("change_language", (_) => {
    if (!select.vluey);

    link = link.replace(register, `${selected.vluey}`);
});
});

```

Б.6 Файл headline.js

```

var firstName = document.querySelector('#userFName');
var lastName = document.querySelector('#userLName');
var userPassword = document.querySelector('#userPass');
var birthDate = document.querySelector('#userBDay');
var form = document.querySelector('#form-Add-User');
var table = document.querySelector('#mainTable');
var btnUpdate = document.querySelector("#btn-update");
form.style.display = 'none';

userPassword.addEventListener('input',() => {
    userPassword.value = userPassword.value.replace(/[а-я]/gi, '');
});

GetAllUsersRequest(table);

var btnRegNewUser = document.querySelector('#btn-submit');
btnRegNewUser.addEventListener('click',()=>{
    formValidate();
    let type = "POST";
    let url = "http://localhost:56481/api/Users";

    AddUserRequest(type,url);
});

var btnAddUser = document.querySelector("#btn-sign-in");
btnAddUser.addEventListener('click',() =>{
    form.style.display = 'block';
    btnRegNewUser.style.display = 'block';

```

```

        btnUpdate.style.display = 'none';
    });

    var appExit = document.querySelector("#btn-exit");
    appExit.addEventListener('click', () =>{
        table.style.display = 'none';
        form.style.display = 'none';
        btnRegNewUser.style.display = 'none';
        btnUpdate.style.display = 'none';
    });

    function formValidate(){
        firstName.maxLength = 20;
        lastName.maxLength = 20;
        userPassword.maxLength = 20;

        let firstNameInput = { value: firstName.value, size: 5, label:
document.querySelector("#lblFname"), span: document.querySelector("#spanFname")};
        let lastNameInput = { value: lastName.value, size: 5, label:
document.querySelector("#lblLname"), span: document.querySelector("#spanLname")}
        let PasswordInput = { value: userPassword.value, size: 8, label:
document.querySelector("#lblPass"), span: document.querySelector("#spanPass")};

        let inputs = [firstNameInput, lastNameInput, PasswordInput];

        inputs.forEach(function(obj, index){

            if(obj.value.length < obj.size){
                obj.span.style.color = "red";
                let text = (`✘ ${obj.label.innerHTML} должен быть длинее
${obj.size-1} символов!`);
                obj.span.innerHTML = text;
            }
            else{
                obj.span.innerHTML = '✓';
                obj.span.style.color = "green";
            }
        });
    }
}

```

```

function createRequestBody(){
    let spanFirstName = document.querySelector('#spanFname');
    let spanLastName = document.querySelector('#spanLname');
    let spanPassword = document.querySelector('#spanPass');
    if(spanFirstName.innerHTML == '√' && spanLastName.innerHTML == '√' &&
spanPassword.innerHTML == '√' ){

        let registered = new Date();
        let picture = null;
        let body = JSON.stringify({
            firstName: firstName.value,
            lastName: lastName.value,
            password: userPassword.value,
            birthDate: birthDate.value,
            registered: registered,
            picture: picture
        });
        window.location.reload();
        form.style.display = 'none';
        return body;
    }
}

```

```

function AddUserRequest(type,url){
    formValidate();
    let body = createRequestBody();
    if (body != null){
        let xhr = new XMLHttpRequest();
        xhr.open(type, url);
        xhr.setRequestHeader('Content-Type','application/json');
        xhr.send([body]);
    }
}

```

```

function UpdateUserRequest(type, url){
    formValidate();
    let body = createRequestBody();
    if (body != null){
        let xhr = new XMLHttpRequest();

```



```

    xhr.open(type, url);
    xhr.setRequestHeader('Content-Type', 'application/json');
    xhr.send([body]);
  }
}

```

```

function DeleteUserRequest(type,url){
  let xhr = new XMLHttpRequest();
  xhr.open(type, url);
  xhr.setRequestHeader('Content-Type', 'application/json');
  xhr.send();
}

```

```

function GetAllUsersRequest(){
  var xhr = new XMLHttpRequest();

  xhr.open("GET", "http://localhost:56481/api/Users");
  xhr.send();
  xhr.onload=function(){
    var users=JSON.parse(xhr.response);

    users.forEach(function(obj, index){
      let buttonRemove = document.createElement("input");
      buttonRemove.type = "button";
      buttonRemove.value = "Remove";
      let buttonUpdate = document.createElement("input");
      buttonUpdate.value = "Update";
      buttonUpdate.type = "button";
      let tr = document.createElement('tr');
      let tdForBtnRemove = document.createElement('td');
      let tdForBtnEdit = document.createElement('td');
      for (let key in obj){
        var td = document.createElement('td');

        tdForBtnRemove.appendChild(buttonRemove);

        tdForBtnEdit.appendChild(buttonUpdate);
        td.innerHTML = obj[key];
        tr.appendChild(td);

        tr.appendChild(tdForBtnEdit);
      }
    });
  }
}

```

```

        tr.append(tdForBtnRemove);
    }
    buttonRemove.addEventListener('click',()=>{
        let tr = buttonRemove.parentNode.parentNode;
        let td = tr.getElementsByTagName("td")[0].innerHTML;
        let userID = td;
        let type = "DELETE";
        let url = `http://localhost:56481/api/Users/${userID}`;
        DeleteUserRequest(type, url);
        window.location.reload();
    });
    buttonUpdate.addEventListener('click',()=>{
        form.style.display = 'block';
        btnRegNewUser.style.display = 'none';
        btnUpdate.style.display = 'block';
        btnUpdate.addEventListener('click',()=>{
            let tr = buttonUpdate.parentNode.parentNode;
            let td = tr.getElementsByTagName("td")[0].innerHTML;
            let userID = td;
            let type = "PUT";
            let url = `http://localhost:56481/api/Users/${userID}`;
            DeleteUserRequest(userID);
            UpdateUserRequest(type, url);
            window.location.reload();
        });
    });

    table.appendChild(tr);
});
}
}

```

ДОДАТОК В
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЗАПОРІЗЬКА ПОЛІТЕХНІКА”

Кафедра програмних засобів

Випускна кваліфікаційна робота бакалавра на тему:

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ГЕНЕРАЦІЇ РЕЗЮМЕ

Виконав

Ст. гр. КНТ-228

Д.А. Сташук

Керівник

к.т.н., доцент

Т.В. Федорончак

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

- Об'єкт дослідження – програмні засоби для створення резюме.
- Предмет дослідження – вебзастосунок, що надає користувачам можливість створювати резюме.
- Мета роботи – програмна реалізація вебзастосунку для створення та завантаження резюме.

Рисунок В.2 – Слайд 2

АНАЛОГИ РОЗРОБЛЮВАНОГО ПРОДУКТУ В МЕРЕЖІ

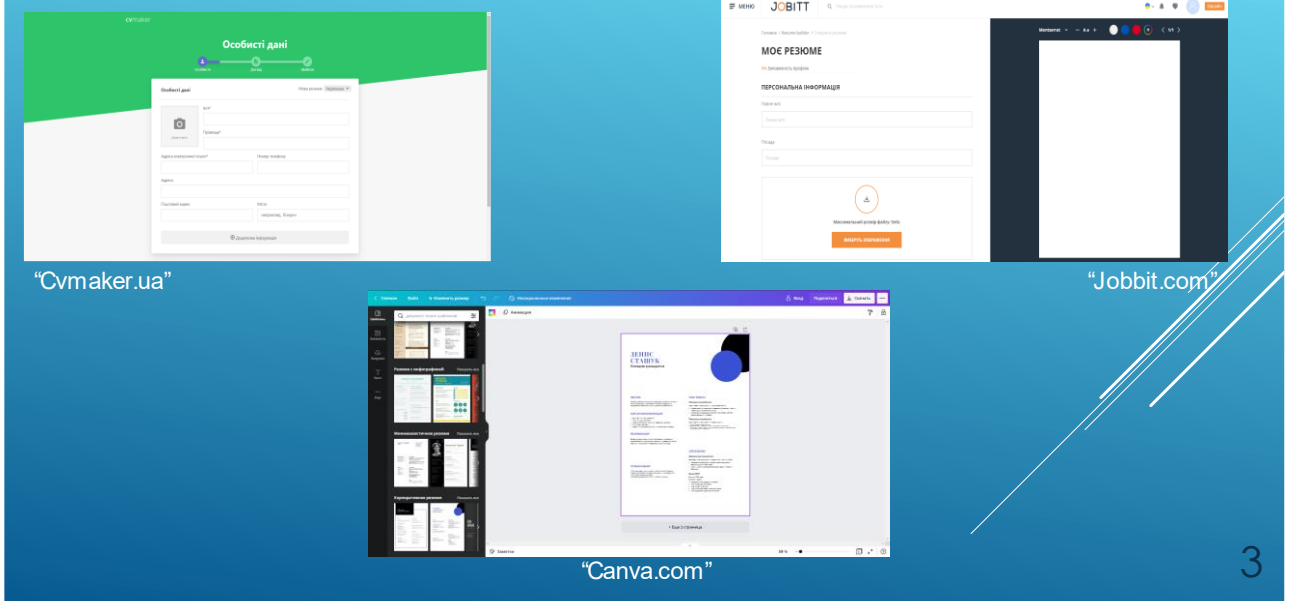


Рисунок В.3 – Слайд 3

ПОРІВНЯЛЬНА ТАБЛИЦЯ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ

	“Cvmaker.ua”	“Jobbit.com”	“Canva.com”
Виконання основного функціоналу (створення резюме та його завантаження)	+	+	+
Адаптивність	+	+	-
Зручний інтерфейс	+	+	-
Платність	+	-	-
Підтримка української мови	+	+	+
Швидкодія	-	+	+
Перегляд резюме у реальному часі	-	+	+

4

Рисунок В.4 – Слайд 4

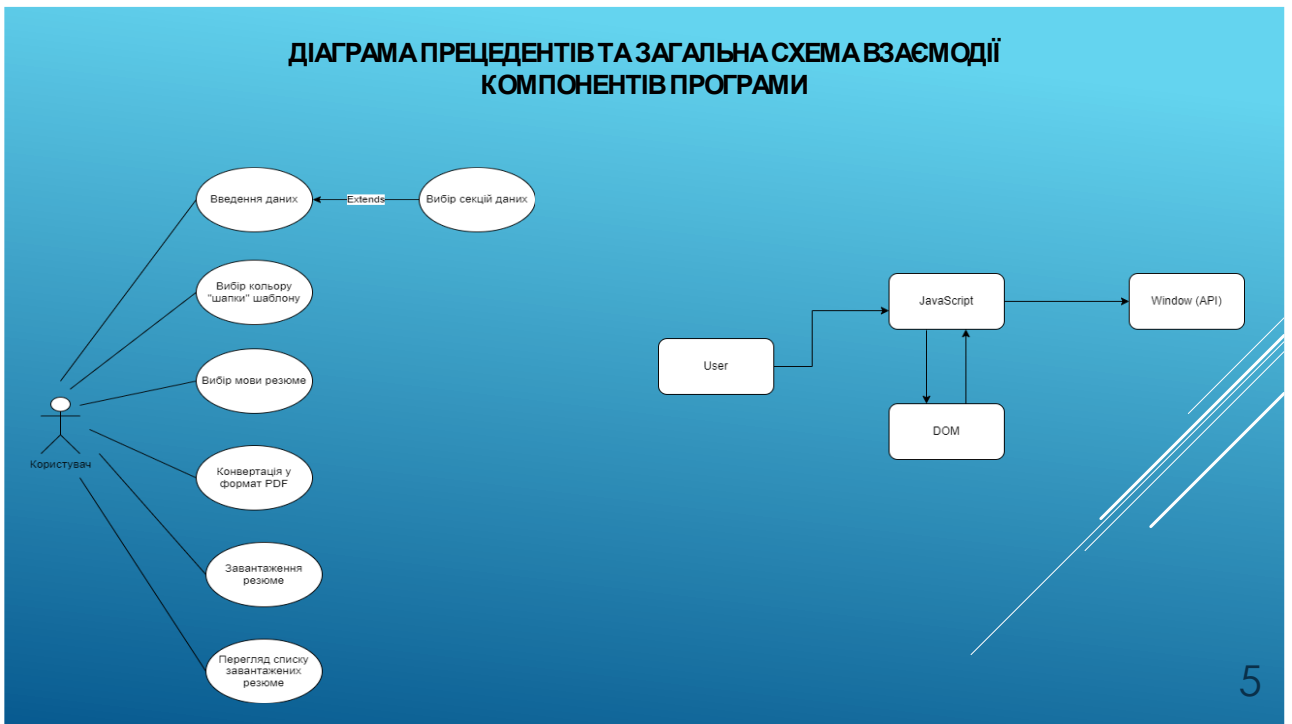
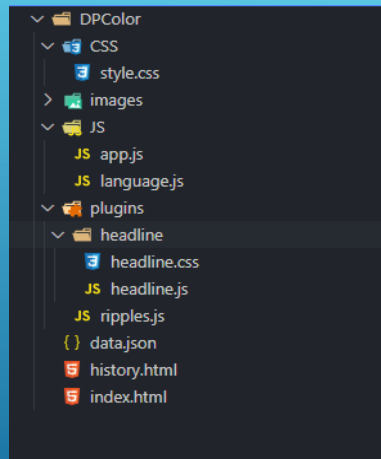


Рисунок В.5 – Слайд 5



Рисунок В.6 – Слайд 6

СТРУКТУРА ФАЙЛІВ ПРОЄКТУ



7

Рисунок В.7 – Слайд 7

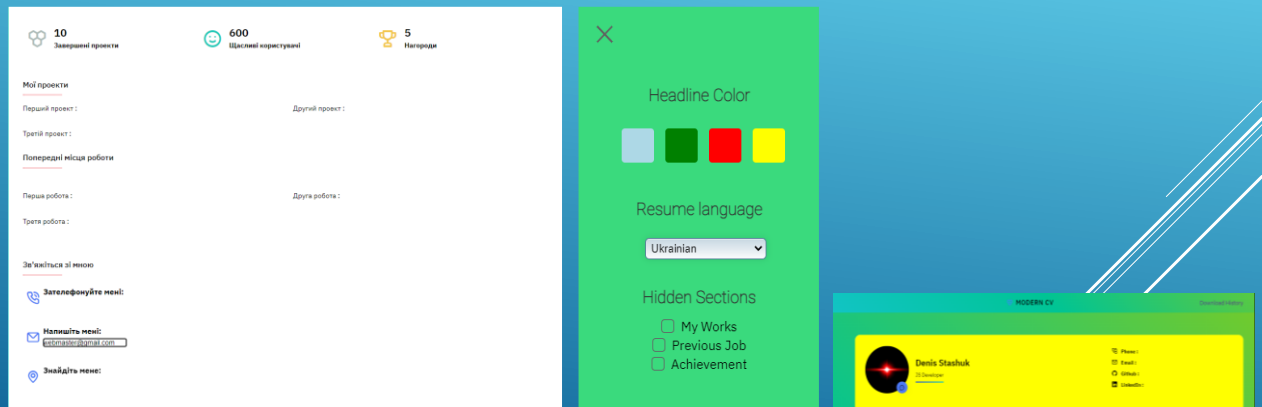
ФУНКЦІЇ РОЗРОБЛЕНОЇ ПРОГРАМИ

- Зміна кольорової теми шапки шаблону
- Вибір мови створення резюме
- Можливість приховувати певні секції
- Сторінка історії завантажених резюме

8

Рисунок В.8 – Слайд 8

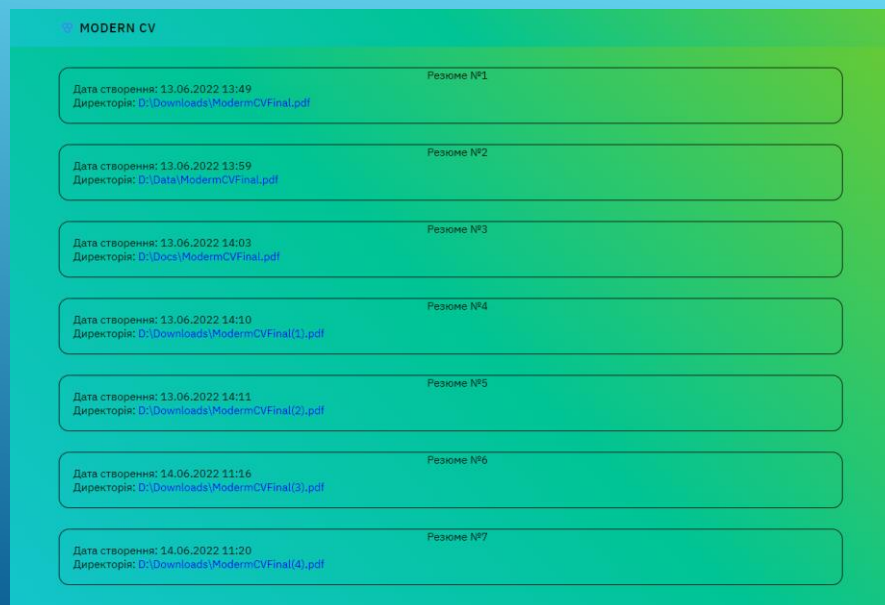
ФУНКЦІОНАЛЬНЕ БУРГЕР-МЕНЮ, ЗМІНА КОЛЬОРУ ШАПКИ ШАБЛОНУ ТА МОВИ СТВОРЕННЯ РЕЗЮМЕ



9

Рисунок В.9 – Слайд 9

СТОРІНКА ІСТОРІЇ ЗАВАНТАЖЕНЬ СТВОРЕНИХ РЕЗЮМЕ



10

Рисунок В.10 – Слайд 10

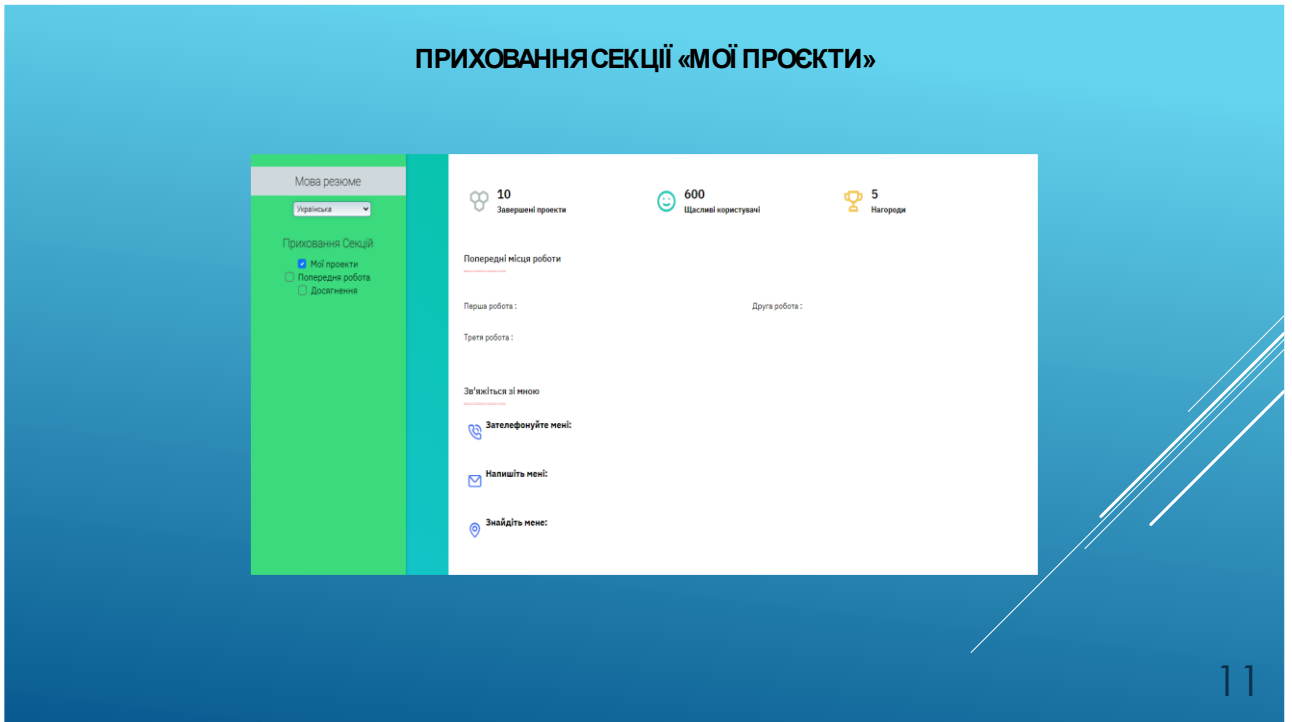


Рисунок В.11 – Слайд 11

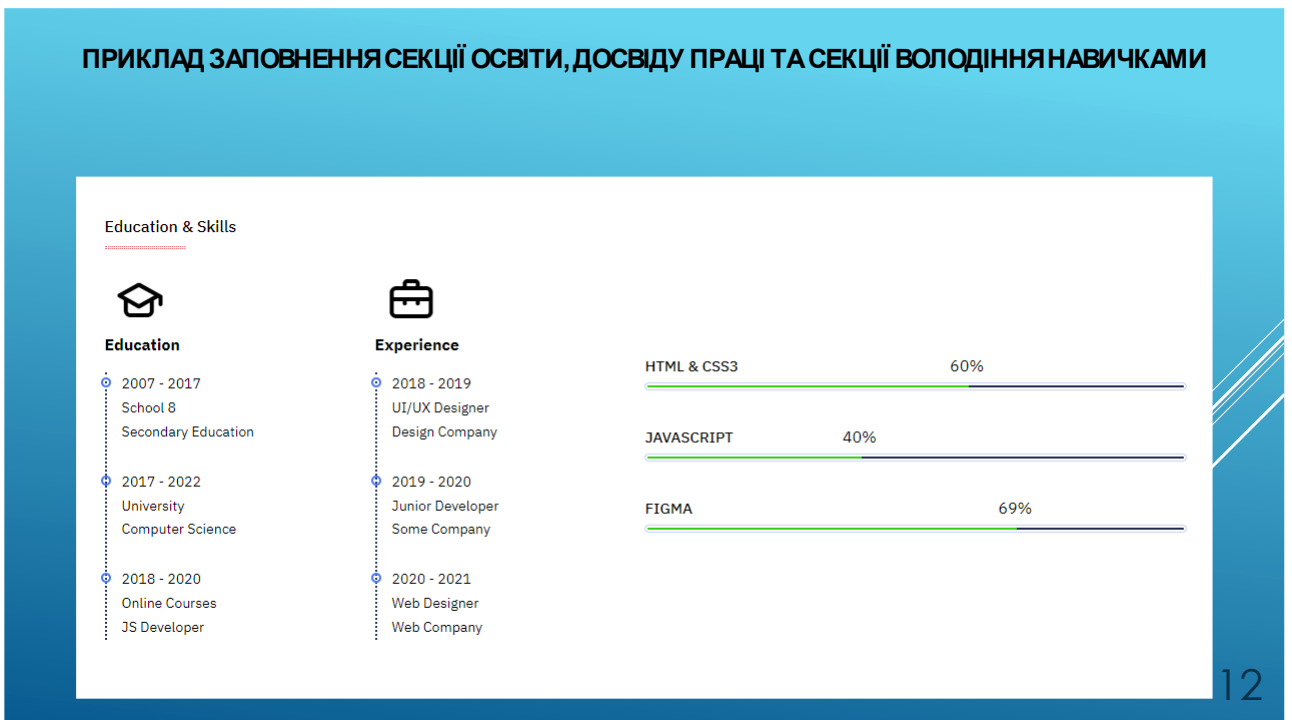


Рисунок В.12 – Слайд 12

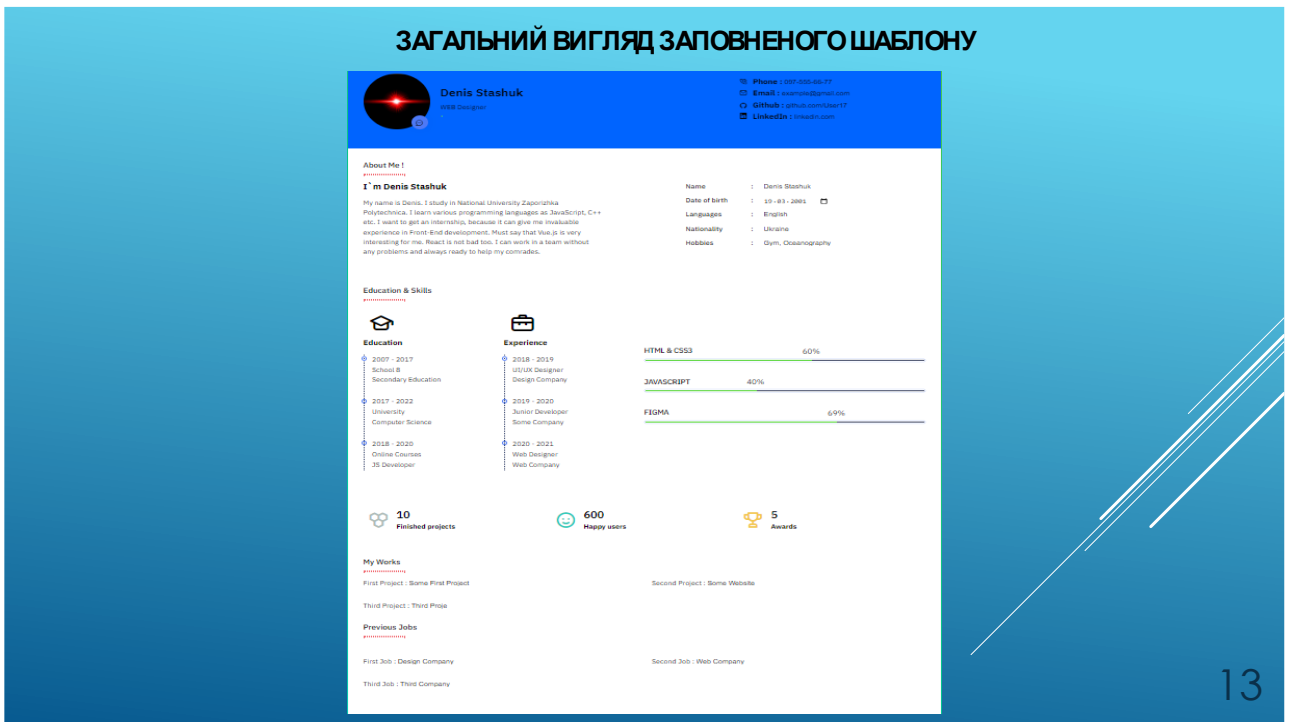


Рисунок В.13 – Слайд 13

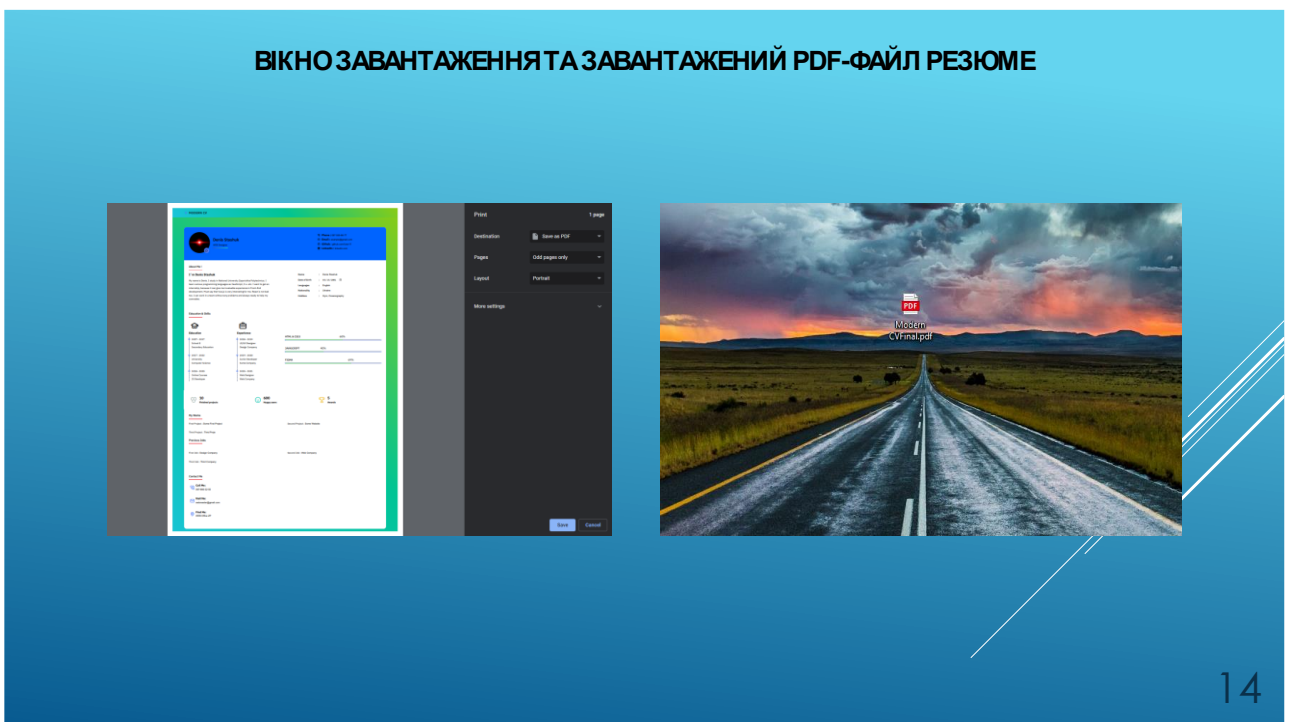
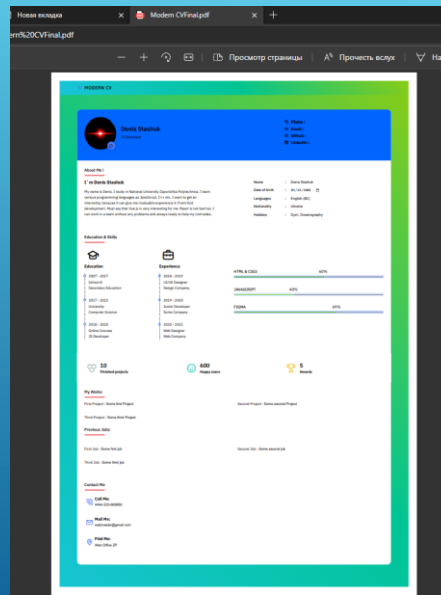


Рисунок В.14 – Слайд 14

ВІДКРИТИЙ У БРАУЗЕРІ PDF-ФАЙЛ РЕЗЮМЕ



15

Рисунок В.15 – Слайд 15

ТЕСТУВАННЯ ЗАСТОСУНКУ В КРИТИЧНИХ СЕРЕДОВИЩАХ

Системні характеристики	Windows 10	Windows 8.1	Windows 7
Оперативна пам'ять 4 Гб Процесор 2 ядра	9.0 секунд	8.5 секунд	8.1 секунд
Оперативна пам'ять 8 Гб Процесор 2 ядра	8.5 секунд	8.5 секунд	7.0 секунд
Оперативна пам'ять 4 Гб Процесор 4 ядра	8.0 секунд	7.5 секунд	7.0 секунд
Оперативна пам'ять 8 Гб Процесор 4 ядра	7.5 секунд	7.3 секунд	6.5 секунд

16

Рисунок В.16 – Слайд 16

ТЕСТУВАННЯ АДАПТИВНОСТІ ВЕБ-ЗАСТОСУНКУ

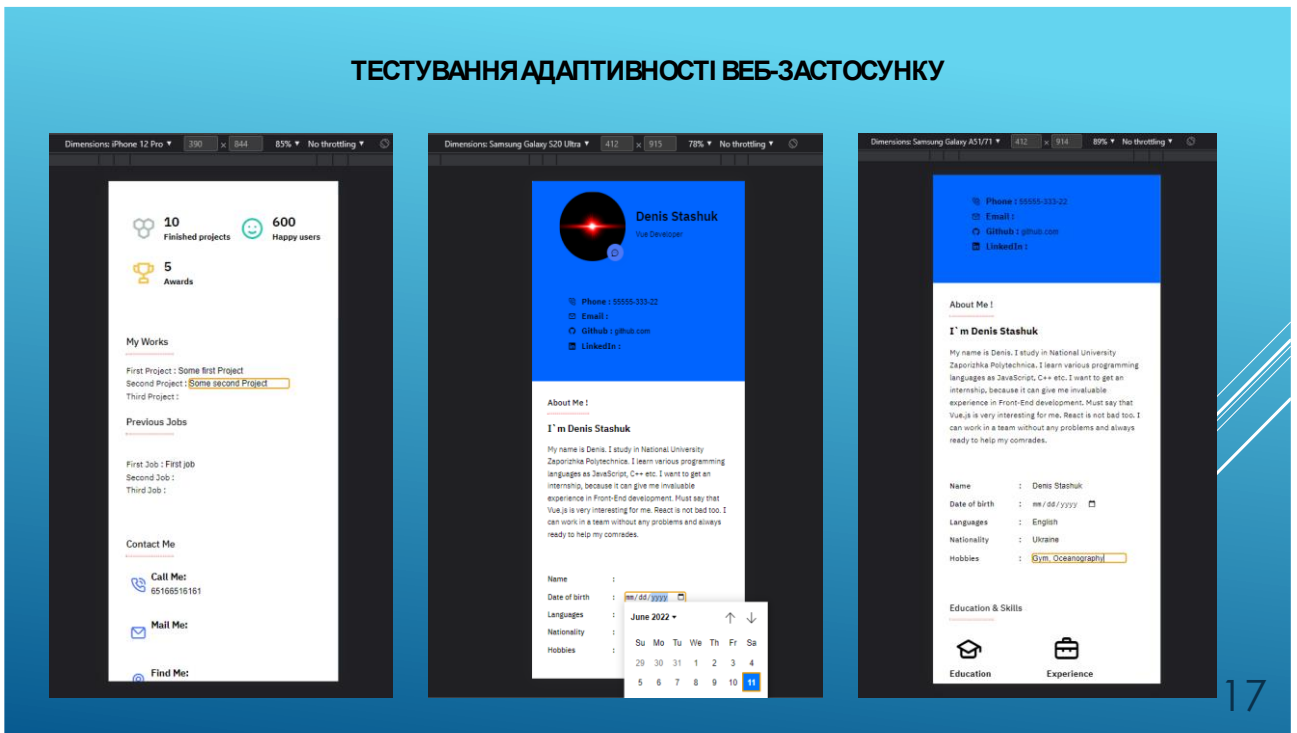


Рисунок В.17 – Слайд 17

ВИСНОВОК

- Під час виконання дипломної кваліфікаційної роботи бакалавра було розроблено вебзастосунок для створення та завантаження резюме "MODERN CV".
- Після проведення аналізу існуючих аналогів було створено вебзастосунок, який, на думку автора, виправляє недоліки аналогів.
- Програму "MODERN CV" можна використовувати для створення персонального резюме.
- Проаналізовано сучасні технології у сфері створення програмного забезпечення. Найпопулярнішим стилем оформлення є мінімалістичний дизайн, який виключає скупчення та нагромадження зайвих, великих та непотрібних об'єктів.

Рисунок В.18 – Слайд 18

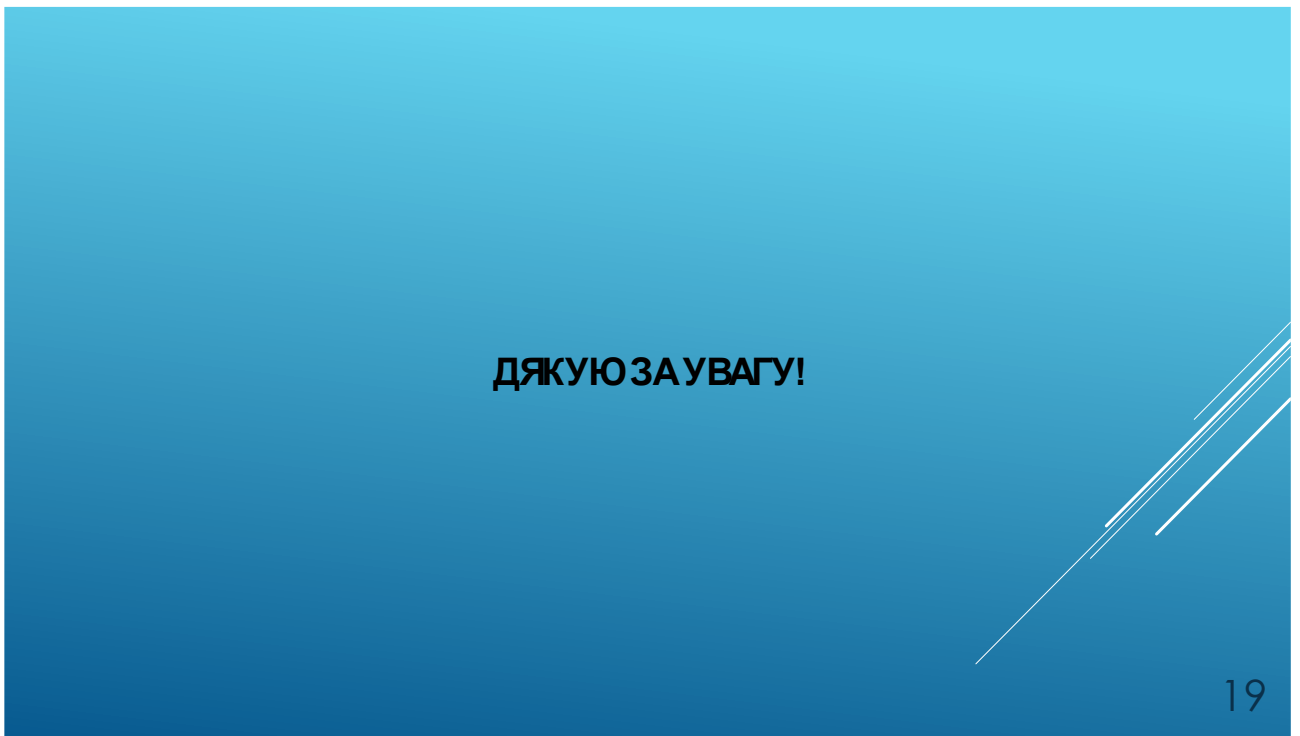


Рисунок В.19 – Слайд 19