

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота
на здобуття ступеня бакалавра
спеціальності 122 «Комп'ютерні науки»

на тему:

**Розробка системи підтримки процесу
працевлаштування**

Виконав студент 4-го курсу
Постільняк Денис Вікторович

Науковий керівник:
к. ф.-м. н., доцент
Панченко Тарас Володимирович



(підпис)



(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до
захисту на засіданні кафедри теорії та
технології програмування

« 01 » червня 2022 р., протокол № 10

Завідувач кафедри

М. С. Нікітченко



(підпис)

РЕФЕРАТ

Обсяг роботи 43 сторінки, 13 ілюстрацій, 2 таблиці, 13 джерел посилань.

TYPESCRIPT, JAVASCRIPT, CSS, HTML, AUTH0,
ВЕБЗАСТОСУНОК

Об'єктом роботи є процес проходження тестів, разом з записом відео, перевіркою камери та мікрофону. Предметом роботи є вебзастосунок для забезпечення можливості аналізу мовлення, зіставлення проаналізованого з правильними відповідями та перегляд записаного проходження тесту для подальшого аналізу.

Метою кваліфікаційної роботи є розробка вебзастосунку «EasyEmployAssistance», який дає змогу перевірки обладнання та проходження тесту, для його подальшого аналізу мовлення та відповідей.

Методи розробки: розробка програмного продукту на основі аналізу мовлення та зіставлених відповідей. Інструменти розробки: інтегроване середовище розробки Visual Studio Code, фреймворк Angular, мова програмування TypeScript

Результати роботи: проведено дослідження основних етапів розробки вебзастосунку та технологій для його написання. Проаналізовано, розроблено, удосконалено та протестовано вебзастосунок «EasyEmployAssistance» для проходження тестів та аналізу результатів.

Вебсайт «EasyEmployAssistance» може застосовуватися для перевірки камери та мікрофону, проходження тестів, розпізнавання мовлення та аналізу відповідей на тест шляхом подальшого зіставлення до правильних відповідей.

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ ЗАСТОСУНКІВ	10
1.1 Вебсайт «Cammiо».....	10
1.2 Вебсайт «Modern Hire»	11
1.3 Вебсайт «Indivizo»	12
1.4 Порівняння розглянутих рішень	13
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	16
2.1 Вибір мови програмування та фреймворку фронтенд частини.....	16
2.2 Вибір технології автентифікації	20
2.3 Вибір технології розпізнавання голосу.....	22
2.4 Вибір технології зіставлення відповідей	23
РОЗДІЛ 3. ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ	24
3.1 Призначення застосунку.....	24
3.2 Цілі розробки застосунку	25
3.3 Вимоги до застосунку	26
3.3.1 Загальні вимоги до застосунку	26
3.3.2 Вимоги до функцій, які виконуються програмою	26
3.3.3 Технічні вимоги до застосунку	28
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ	29
4.1 Розробка вебзастосунку.....	29
4.2 Постановка завдання.....	29
4.3 Структура програми.....	30

	4
4.3.1 Клієнтська частина	30
4.3.2 Інтеграція зі стороннім сервісом	31
4.4 Тестування сайту	32
4.4.1 Модульне тестування.....	33
РОЗДІЛ 5. ІНСТРУКЦІЯ КОРИСТУВАЧА	34
ВИСНОВКИ	41
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – Application Programming Interface;

HTTP – HyperText Transfer Protocol;

CSS – Cascading Style Sheets;

SCSS – Syntactically Cascading Style Sheets

HTML – HyperText Markup Language;

JS – JavaScript;

TS – TypeScript;

JSON – JavaScript Object Notation;

IDE – Integrated Development Environment;

IT – Information Technology ;

CRM – Customer Relationship Management;

SSO – Single Sign On;

SPA – Single Page Application;

MFA - Multifactor authentication;

DI – Dependency Injection;

CV - Curriculum vitae;

Рекрутинг – процес підбору персоналу;

Фіча – особливість;

ВСТУП

Оцінка сучасного стану об'єкта розробки. Останнім часом світ все більше і більше починає діджиталізуватися: компанії трансформують свій бізнес в онлайн середовища, де є нова клієнтська база, нові продукти та нові цікаві дороги розвитку; сфера послуг також не відстає і клієнти все більше і більше вимагають сучасних рішень для підвищення комфорту, збереження часу, зменшення витрат і тд. Тому зараз одним із найпопулярніших напрямлень є ІТ.

Ринок ІТ перегрітий, вакансій все більше і більше, а розробників не так і багато, і це є проблемою. Багато ІТ компаній наймають до себе рекрутерів, але деякі користуються послугами агенцій пошуку кандидатів, які так само наймають рекрутерів і всі ці витримки кладуться на саму компанію. В наш час знаходити кандидата є затратно не тільки по часу але й по коштам. Тому для мінімізації цих витрат придумано дуже багато рішень, які можуть полегшити життя кандидату, рекрутеру та людині яка проводить інтерв'ю або задіяна в цьому процесі.

Мені на думку прийшла ідея реалізації такого продукту, який би зміг полегшити цей процес та допомогти компаніям зменшити витрати. Зробивши глибокий аналіз пропозицій на ринку я зрозумів, що всі продукти пропонують більше CRM рішення, які допомагають вести облік кандидатів, вакансій, та в загальному процес рекрутингу кандидата.

Тому було прийняте рішення розробки системи допомоги рекрутингу, який би зміг проводити автоматичне інтерв'ю з кандидатом. Цей застосунок багатофункціональний: в нього додано основні, найбільш популярні напрямки сьогодення.

Одне з головних завдань цієї системи – це відповідь кандидата на запитання, шляхом запису відео з камери та голосу через мікрофон. Система

допомагає розпізнати текст з голосу, цим самим вирішити питання часу, та взаємодії з кандидатом, а в комбінації з відеозаписом зменшити вірогідність якихось шахрайських дій. Аналіз голосу може розпізнати сказаний текст, і записати його як відповідь. В наші часи ця тематика є дуже популярною, тому робота з цією технологією є дуже корисною та цікавою.

Також є можливість порівнювати розпізнану відповідь з правильною відповіддю та пошуку відсотку сходження. Цей пошук відбувається за допомогою алгоритму Левенштейнова відстань. Відсоток грає важливу роль в подальшому аналізі кандидата, його soft та hard skills.

Голос – дуже потужний інструмент, за допомогою якого можна визначити моральний та психологічний стан людини, її настрій, та інші параметри. Також можна дослідити чистоту та рівень мовлення людини в відповідній мові. Разом з відеозаписом можна зіставити картину того, як кандидат проходить інтерв'ю.

Актуальність роботи. Для того щоб компаніям менше витратити коштів і часу на процес рекрутингу, а кандидатам було цікавіше і легше проводити інтерв'ю в меншій кількості етапів, потрібно використовувати відповідні інструменти для цього. Деякі дослідження доводять, що правильне використання налаштованих програмних продуктів підвищує продуктивність.

Кандидат за дуже малу кількість часу може зайти в веб застосунок і пройти будь-який потрібний тест без зайвих зусиль та попередніх планувань часу. Цей привілей дає змогу не носити з собою постійно електронні пристрої з можливістю пройти інтерв'ю.

Важливим критерієм оцінки по праву може вважатися зручність, гнучкість і функціональність у використанні. Сучасні підходи, які допомагають удосконалити та спростити процес рекрутингу – важлива перевага запропонованого в кваліфікаційній роботі застосунку.

Мета й завдання роботи. Метою роботи є розробка вебзастосунку «EasyEmployAssistance», який містить систему для проходження інтерв'ю, що дозволяє аналізувати мовлення та співвідношення відповідей. Для досягнення поставленої мети створено такі завдання:

- Дослідити існуючі вебзастосунки для проходження автоматичного інтерв'ю
- Апробувати технології (Annyang, Levenshtein dist.) для проєктування, реалізації веб-сайту та інтеграції з зовнішніми ресурсами (Auth0, Ngx-bootstrap)
- Розробити вебзастосунок «EasyEmployAssistance» для проходження інтерв'ю та його відповідні компоненти

Об'єкт, методи та засоби розроблення. Об'єктом розроблення програми «EasyEmployAssistance» є процес проходження інтерв'ю, збереження отриманих результатів, їх подальший аналіз та зіставлення з правильними відповідями для зображення результату. Перегляд записаного відео для психологічного та емоційного аналізу людини, а також для запобігання шахрайських дій.

Предметом роботи є вебзастосунок для забезпечення можливості проходження інтерв'ю.

Розробці програмного засобу передував аналіз готових рішень проходження інтерв'ю.

Для створення застосунку було використано інструмент Visual Studio Code – автономний редактор коду мовою програмування TS від компанії Microsoft.

Можливі сфери застосування. Вебзастосунок «EasyEmployAssistance» може бути застосований в сфері ІТ, а саме для компаній, які в пошуках

кандидата та які хочуть зберегти купу ресурсів. Також застосунок може бути використаний окремо для кандидатів для перевірки soft and hard skills.

РОЗДІЛ 1.

ОГЛЯД НАЯВНИХ НА РИНКУ ЗАСТОСУНКІВ

В наш час можна знайти невелику кількість вебзастосунків для проходження автоматичного інтерв'ю. Розглянемо детальніше найпопулярніші з них, знайдемо переваги, недоліки та місця покращення.

1.1 Вебсайт «Самміо»

Одну з передових позицій в переліку найбільш популярних застосунків для проходження автоматичного онлайн інтерв'ю займає «Самміо». Власники стверджують, що відео рекрутинг все швидше і швидше стає нормою в скринінгу та взаємодією з кандидатом, як для роботодавця так і для самого кандидата. Якщо вдаватися в цифри то цей застосунок зберігає дуже багато ресурсів, а саме: щонайменше 25% зменшення фінансових витрат на процес рекрутингу, до 60% швидше знаходиться кандидат та 100% кращі відгуки від самих кандидатів по закінченню цього процесу[6].

З основних фіч цього застосунку є вступний ролик. Кандидат розповідає про свою мотивацію в короткому ролику та додає щось саме основне про себе. Найчастіше така фіча використовується як доповнення до CV на стадії пре-скріну. Цей процес забезпечує перше візуальне враження та дає глибше зрозуміти мотивацію кандидата.

Далі йде автоматичне інтерв'ю під час якого кандидат відповідає на наперед записані питання з камерою від інтерв'ювера. Цей етап найчастіше заміняє дзвінок по телефону, частіше для позицій entry-middle. Допомогає проскрінити більше кандидатів за менший проміжок часу забезпечуючи однаковий процес для всіх. Зображення процесів наведено на рис.1.

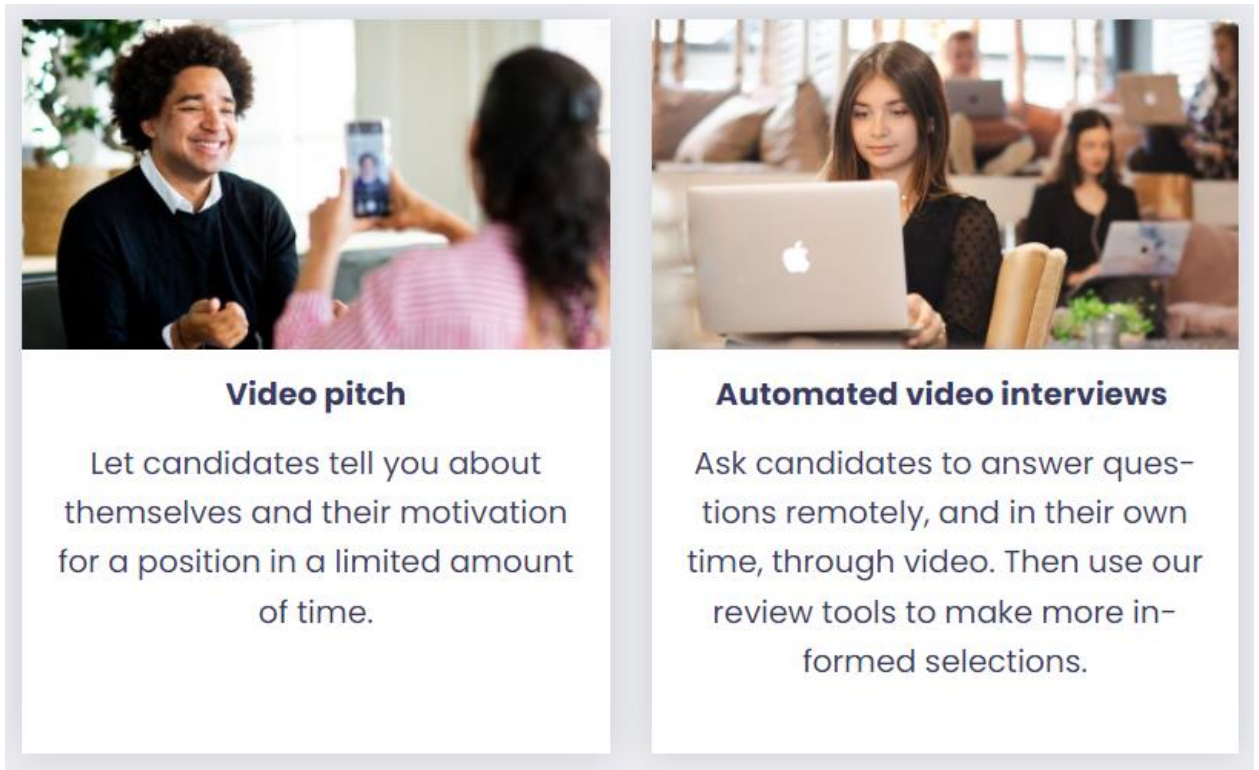


Рисунок 1. Основні фічі програми «Cammio»

1.2 Вебсайт «Modern Hire»

«Modern Hire» - це потужний вебзастосунок для кастомізації проходження онлайн інтерв'ю. В даному застосунку запитання сприймаються кандидатом не з наперед записаного відео ролику, а з текстових запитань. Потужний модуль з великою базою даних питань, дозволяє інтерв'юверу вибрати найбільш підходящі питання для конкретної компетенції, що дозволить бути біль точним та вибрати правильного кандидата[5].

Програма дає можливість використати автоматичний асистент для взаємодії з кандидатом. Можна додати кандидата до відповідної групи розсилки, далі налаштувати метод контакту: текстове або email повідомлення, система зі штучним інтелектом яка буде займатися розсилкою. Кандидати самі вибирають зручний час та день, не прив'язуючись до когось.

З провідних фішок цієї системи є також автоматичний підрахунок набраних результатів. Цей модуль аналізує відповіді кандидата, запити по відповідній вакансії, та зображує відповідний набраний бал, який говорить на скільки людина підходить даній вакансії. Система може запропонувати інші відкриті вакансії, якщо ті більше будуть підходити під компетентності кандидата. Приклад користувацького інтерфейсу наведено на рис.2.

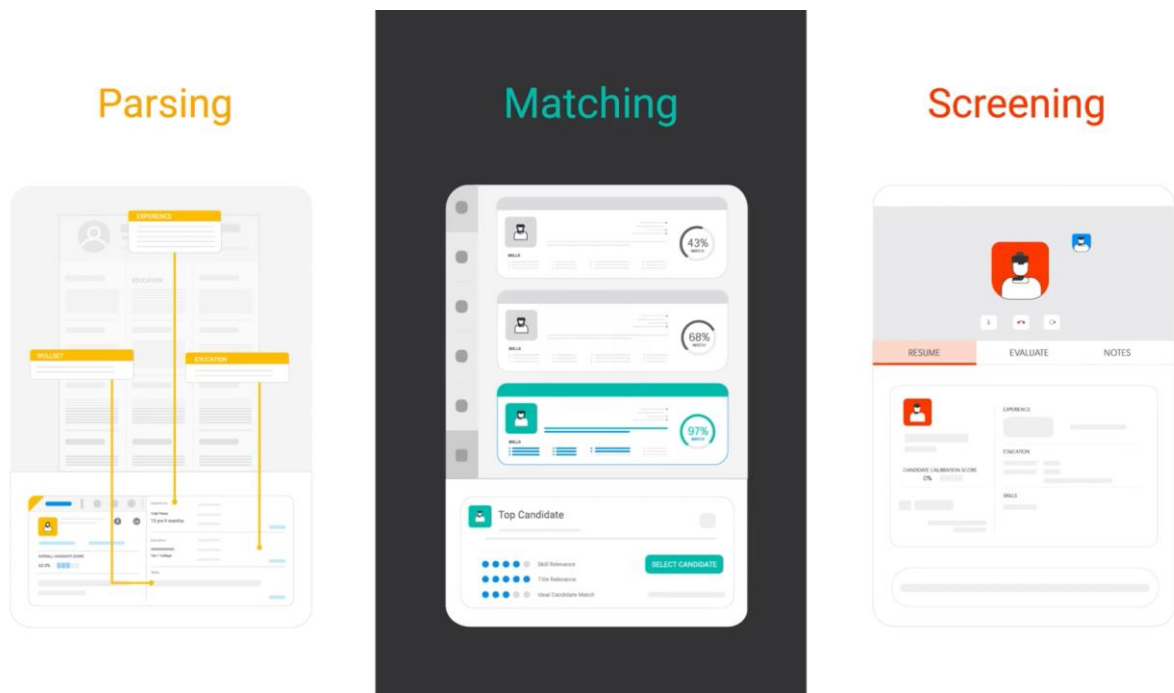


Рисунок 2. Сторінки з основними функціями програми «Modern Hire»

1.3 Вебсайт «Indivizo»

Цей застосунок дозволяє проходити автоматичні відео інтерв'ю за допомогою записаних коротких роликів. Так само тут можна створити свої власні вакансії або розділи, і вже в відповідні частини створювати запитання, давати час проходження і інші властивості[7].

Функціонал більш адаптований та розвинений під інтерв'юверів, які оцінюють кандидатів. Людина може вибрати вакансію, далі може переглянути відгуки кандидатів на відповідну вакансію, а потім перейти до перегляду. Для кожного кандидата є відповідні шкали для оцінювання. Можна змінити автоматично вибрані результати, якщо на думку спаде що система оцінила неправильно. Також є функція оцінки за допомогою команди, кожен учасник якої може висловити свої технічні та комунікаційні враження від кандидата, понизити або повисити рейтинг. Приклад користувацького інтерфейсу вебсайту «Indivizo» рис. 3.

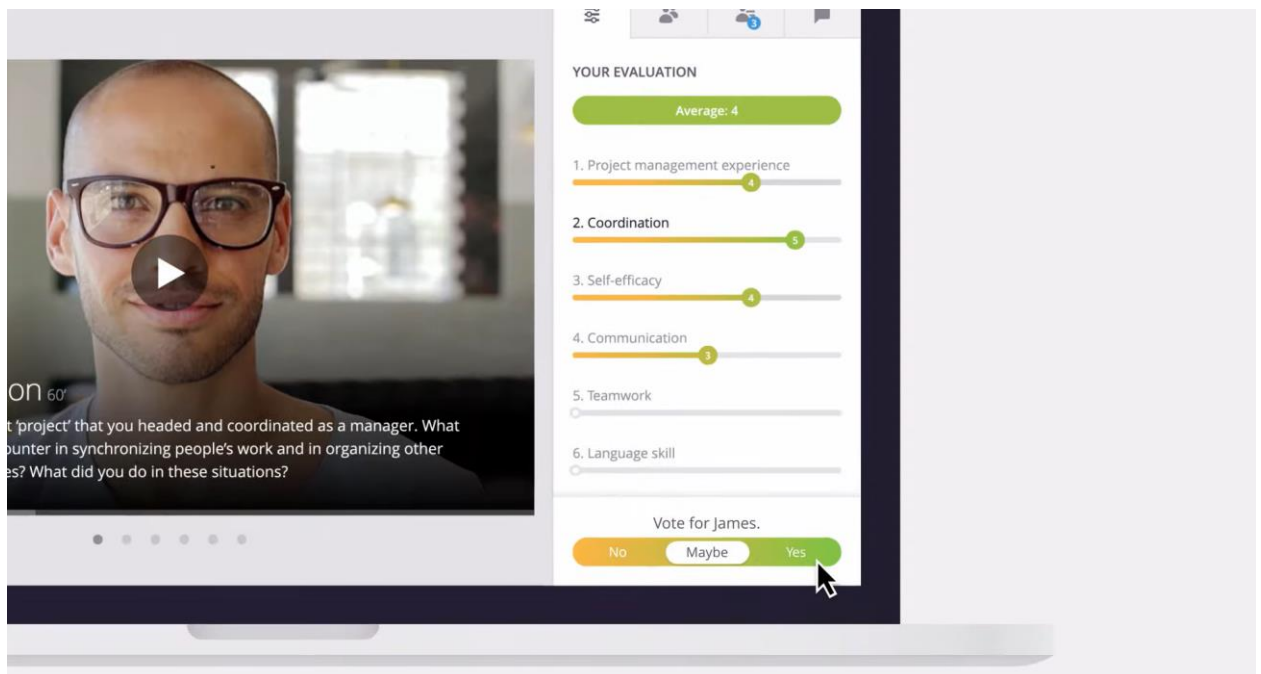


Рисунок 3. Сторінка оцінювання кандидата «Indivizo»

1.4 Порівняння розглянутих рішень

Деякі елементи розглянутих вище застосунків співпадають, хоча більшість з них все ж таки відрізняються незважаючи на їх цільове призначення. Так як ми не маємо інформації про технологічні рішення застосунків, можна зробити порівняння на основі зручності для користувача,

базового функціоналу, додаткових можливостей, досвід використання, інтеграції з іншими сервісами, функціонал для рекрутерів (див. табл. 1).

Таблиця 1. Порівняння сервісів

Назва	Сammio	Modern Hire	Indivizo
Зручність для користувача	1	0.5	0.5
Базовий функціонал	1	1	1
Додаткові можливості	0.5	1	0.5
Досвід використання	1	0.5	1
Інтеграція з іншими сервісами	0	1	0.5
Функціонал для рекрутерів	0.5	1	1

Проаналізувавши відповідні характеристики, наведені в табл. 1 можна зробити висновок, що найкраще виглядає застосунок «Modern Hire», навіть незважаючи на деякі незручності в інтерфейсі. Тому при розробці власної реалізації буде зроблений акцент на подібні варіанти застосунків, але будуть враховані все недоліки.

На основі аналізу сайтів проходження автоматичного інтерв'ю, що були розглянуті вище, можна стверджувати, що розробка вебзастосунку, який

повинен поєднати всі переваги кожного з них та усунути їх недоліки є доцільною.

РОЗДІЛ 2.

ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

Розглянемо інструменти, технології, зовнішні сервіси та рішення, які допоможуть виконати раніше поставлені задачі та цілі найефективнішим способом. Було прийнято рішення розглядати та застосовувати найпопулярніші рішення та ресурси через обмеження в терміні створення вебзастосунку.

2.1 Вибір мови програмування та фреймворку фронтенд частини

Найбільш важливою частиною розробки вебзастосунку є проектування архітектури, так як правильно спроектована архітектура стає більш гнучкою та спрощує подальшу підтримку застосунку (див. рис. 4). Допущені помилки, здійснені при проектуванні, потребують великих затрат для виявлення та виправлення на подальших етапах.

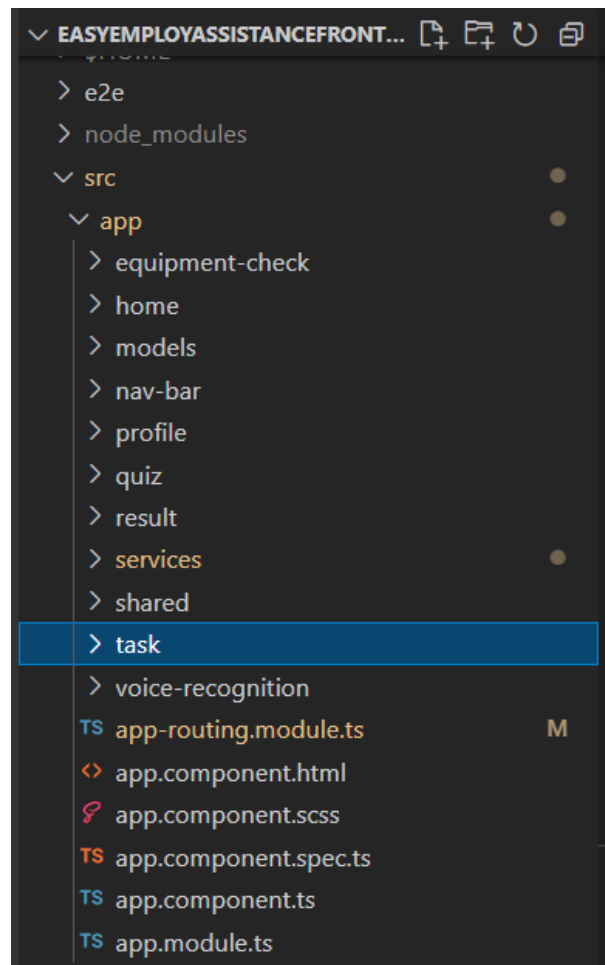


Рисунок 4. Архітектура вебзастосунку

Тому потрібно приділити особливу увагу вибору мови програмування, інструментів та інших ресурсів. Для того, щоб максимально задовольнити вимоги застосунку та виконати основні завдання, необхідно обирати відповідні підходящі технології, адже багатофункціональні можливості фреймворків та бібліотек дозволяють розробнику менше витратити часу та ресурсів на написання власного рішення та використовувати вже готові. Функціонал обраних технологій полегшує розширення застосунку та внесення змін до нього.

Також для того, щоб більш точно відобразити ідею застосунку, та якість задокументувати поетапну розробку – було використано програму для створення мокапів «Mockups». За допомогою цієї програми, можна створити

гарний дизайн застосунку, його каркас, прототип та саме головне, що все документується та всі ідеї не забудуться з часом і потім можна більш точно згадати на чому була зупинена розробка, тобто це дуже корисний інструмент.

Вибір технологій для реалізації вебзастосунку потрібно здійснювати з урахуванням наступних вимог:

- застосунок повинен мати архітектуру яка допоможе спростити розширення та модифікацію, тому обрана мова програмування та фреймворк повинна максимально це зробити;
- застосунок інтегрується з зовнішнім сервісом для автентифікації та авторизації, тому для того щоб максимально зменшити кількість коду та покращити його читабельність, додати можливість легкої заміни провайдера потрібно врахувати це в архітектуру

Frontend частина вебзастосунку – це та частина застосунку, яку користувач бачить та працює з нею. Графічний інтерфейс за допомогою відповідних інструментів відображається в браузері. Взаємодія застосунку з користувачем відбувається через браузер: гортаючи контент, взаємодіючи з активними елементами та вводячи дані.

Angular представляє собою фреймворк, який випустила компанія Google для того, щоб полегшити створення клієнтських програм та покращити вигляд застосунку. Головна його ціль - SPA-рішення, тобто додатків, які фактично використовують одну сторінку. Angular не є першовідкривачем цієї технології, він успадкував це від AngularJS. Але Angular не є продовженням AngularJS або його новою версією, а є взагалі іншим фреймворком[2].

Angular має дуже потужний функціонал, одним з них є двостороннє зв'язування, яке дозволяє пов'язувати код застосунку з HTML частиною, для того щоб постійно відбувалася синхронізація даними.

З особливостей Angular є те, що в ньому використовується мова програмування TypeScript, але це не є обов'язковим. Можна використовувати

різні мови: Dart або JS, але TS є загально прийнятою і найбільш популярною для даного фреймворку [13].

Багато хто задається питанням, навіщо нам якийсь інструмент для роботи з клієнтською частиною, якщо вже існує традиційний JavaScript, який використовується практично на кожному ресурсі, який знає багато розробників і підтримка якого в програмістів висока . Але TypeScript це не заміна JavaScript, а скоріш його оболонка.

По-перше, слід сказати, що TypeScript - це строго типізована і компільована мова, чим, можливо, буде найбільш схожою до мов програмування Java, C#. Але в результаті компілятор все рівно створює JS код з TS, який потім виконується в браузері. Однак ця строга типізація зменшує кількість можливих помилок в коді, які могли б виникнути під час написання на JavaScript та є більш приємним для написання.

По-друге, TypeScript має багато властивостей, які належать до об'єктно-орієнтовних мов такі, як успадкування, поліморфізм, інкапсуляція, абстракція тощо.

По-третє, TypeScript надає розробнику можливість швидше та простіше писати комплексні додатки, як результат їх легше підтримувати, розширювати та тестувати ніж звичайний JS.

По-четверте, TypeScript розвивається як opensource-проект і, як і багато фреймворків, хоститься на гітхабі. Також він є кросплатформний - це означає, що ми не прив'язуємося до конкретної ОС, а можемо використовувати як Windows, так і MacOS або Linux подібні ОС.

Ще не менш важливою особливістю є те, що TypeScript є строго типізованою надмножиною JavaScript, і з цього можна зробити висновок, що JS програма є програмою на TypeScript. TS підтримує всі конструкції, які використовуються в JS – оператори, умовні та циклічні конструкції. Можна зробити висновок, що TS - це лише мова програмування, яка покликана

полегшити розробку додатків. Компілятор TS, який знаходиться в Node.js, можна встановити за допомогою npm.

Для того, щоб написати якусь компоненту мовою TypeScript можна використати самі прості редактори коду починаючи з блокноту і терміналу для компіляції. Більшість редакторів і середовищ розробки, такі як Visual Code Studio, Sublime, Atom, Netbeans, Visual Studio, WebStorm та інші, мають підтримку TypeScript і можна встановити розширення, що дозволять розробнику використати функціональні можливості редактору на максимум.

Мова, якою описується розмітка сторінок вебзастосунку - це HTML. Ця мова описує структуру сторінки, розміщення в ній окремих елементів. Зовнішнє оформлення сторінок, їх стилі і колірна гама описуються в файлі стилів – SCSS.

Для покращення графічного інтерфейсу, надання йому динамічності та кращого зовнішнього вигляду, було використано додаткові ресурси: скрипти JavaScript, бібліотека Bootstrap, Angular Material[9].

2.2 Вибір технології автентифікації

Автентифікація та авторизація необхідні для багатьох програм. Припустимо, ви розробили додаток і впровадили в нього автентифікацію та авторизацію - використовували для цього сторонню бібліотеку автентифікації або платформу ідентифікації. Виконавши свою роботу, як звичайно ви не стали розбиратися, що відбувається за лаштунками, або чому щось відбувається певним чином.

У широкому сенсі, автентифікація відноситься до процесу перевірки того, що користувач є тим, ким він заявляє себе. Авторизація стосується надання чи відмови у правах доступу до ресурсів.

OAuth 2.0 є однією з найбільш часто згадуваних специфікацій, коли йдеться про технологію авторизації та базової автентифікації, а також часто неправильно представленою або неправильно зрозумілою. OAuth не є специфікацією автентифікації. OAuth має справу з делегованою авторизацією, бо автентифікація як ми знаємо – це перевірка особи користувача. Авторизація стосується надання чи відмови у доступі до ресурсів.

Тому мною було прийнято рішення інтегрувати сервіс Auth0 — це досить функціональне рішення для роботи з автентифікацією та авторизацією. Я прийшов до цього рішення для того, щоб уникнути витрат часу, ризиків, пов'язаних із створенням власного рішення та дозволити зручніше автентифікувати та авторизувати користувачів[2].

Користувачі можуть мати можливість входити або за допомогою імені користувача/пароля, або за допомогою своїх облікових записів у соціальних мережах (наприклад, Facebook або Twitter). Можна отримати профіль користувача після входу, щоб можна було налаштувати інтерфейс користувача та застосувати свою політику авторизації. Цей сервіс використовує технологію OAuth 2.0 яка була згадана раніше. Також за можна впровадити єдиний вхід (SSO).

Якщо власник запідозрив, що паролі зламані, то він може запропонувати, щоб його користувачі входили за допомогою одноразових кодів, надісланих електронною поштою або SMS. Ще однією корисною фічею є блокування підозрілих IP-адресів, якщо вони роблять послідовні невдалі спроби входу, щоб уникнути DDoS-атак.

Також однією з популярних фіч є багатofакторна автентифікація (MFA), коли користувачі хочуть отримати доступ до конфіденційних даних. В цьому сервісі є дуже потужна панель управління користувача і в загальному обліковим записом. Із основних: це можливість керувати користувачами,

їхніми активними сесіями, надавати їм ролі і тд. Нижче наведено зображення панелі управління «Auth0» рис. 5.

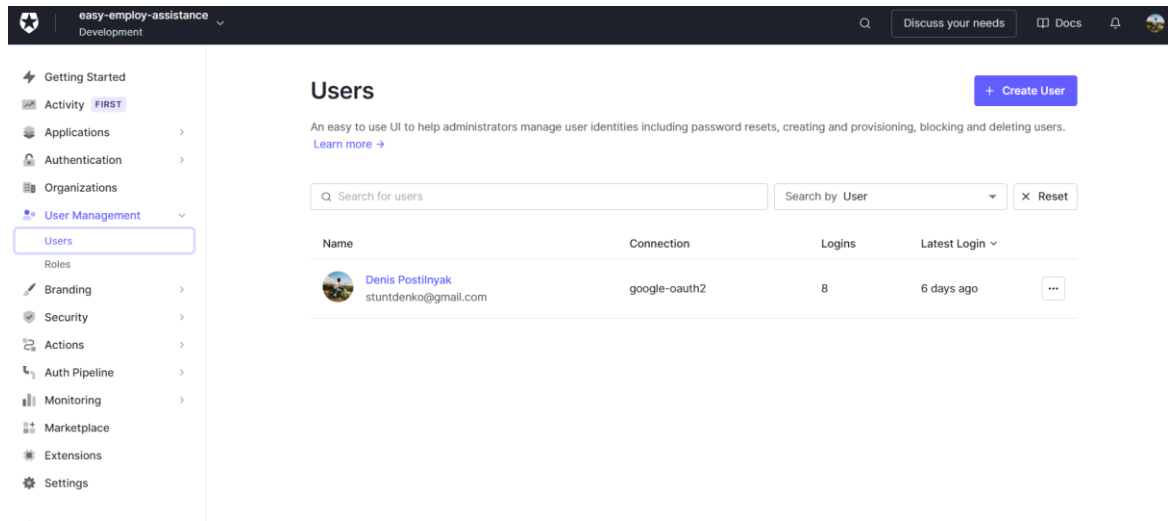


Рисунок 5. Користувацький інтерфейс «Auth0»

2.3 Вибір технології розпізнавання голосу

Для розпізнавання голосу була інтегрована JS бібліотека «annyang». Ця бібліотека є open-source тому в ній багато функціоналу та багато виправлень зі сторони розробників. Можна виділити додаткові властивості цієї бібліотеки[3]:

- Можна промовляти розділові знаки для кращого розуміння та для чіткішого розпізнавання;
- Може типізувати речення по його інтонації або типу(привітання, подяка і тд.);
- Можливість дозволити розпізнавати тільки вимову користувача, яка підходить під regex вираз;
- Підтримка івентів, які ловлять різні типи помилок, найчастіше з проблемою розпізнавання, заборонаю використовувати мікрофон браузером або технічною проблемою;

Ця бібліотека добре працює з усіма браузерами, поступово удосконалюючи браузери які підтримують SpeechRecognition. Бібліотеку легко підключити до будь-якого JS фреймворку, всього в декілька кліків додавши шлях до файлу зі скриптом та викликавши методи цього файлу. Підтримка близько 50-ти мов дає змогу використати цю бібліотеку не залежно від регіону використання, але на жаль української мови в списку немає(але думаю скоро з'явиться).

2.4 Вибір технології зіставлення відповідей

Для того, щоб розпізнану відповідь якимось можна було проаналізувати і зобразити наскільки близькою до правильної відповіді вона була – використано алгоритм «Левенштейнова відстань»[8].

У теорії інформації, лінгвістиці та інформатиці відстань Левенштейна є величиною для вимірювання різниці між двома послідовними значеннями. З інших ресурсів, відстань Левенштейна — це мінімальна кількість односимвольних редагувань (вставки, видалення чи заміни), необхідних для перетворення одного слова на інше.

Тому було реалізовано цей алгоритм, в окремому сервісі створено метод, який приймає дві змінні строкового типу та в результаті повертає їх збіжність. Ці дані зображуються на останньому етапі проходження інтерв'ю, де можна побачити результати по кожному запитанню тесту.

РОЗДІЛ 3.

ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ

3.1 Призначення застосунку

Основним завданням даної роботи є створення системи автоматичного проходження інтерв'ю, з подальшим аналізом відповідей кандидата та зіставленням з правильними. Розробка повинна відповідати наступним основним критеріям:

- надавати зручне та зрозуміле користування застосунком;
- надати можливість користувача безпечно пройти автентифікацію та авторизуватися;
- надати можливість користувачу вибрати бажаний тест та переглянути інформацію по ньому;
- надати можливість перевірки камери та мікрофону на справність;
- надати можливість розпізнати мову користувача;
- надати можливість переглянути проаналізовану інформацію та результати тесту;
- надати можливість переглянути запис проходження інтерв'ю;

Провівши аналіз уже готових рішень та їх призначень, можна зробити висновок, що актуальною буде розробка зі зручним інтерфейсом, простим в використанні, та достатньою кількістю функціоналу. Це забезпечить зрозумілий, функціональний та зручний доступ до застосунку для користувача, оскільки такі системи все більше і більше стають популярними з ростом ІТ.

Отже, очікується отримати багатофункціональний вебсайт з можливістю проходження автоматичного інтерв'ю та його подальшим аналізом.

3.2 Цілі розробки застосунку

Застосунок «EasyEmployAssistance» розробляється з метою:

- забезпечення можливості перевірки камери та мікрофону;
- проходження автоматичного інтерв'ю;
- аналіз мовлення кандидата та його текстове зображення;
- зіставлення проаналізованих відповідей з правильними;
- перегляд записаного інтерв'ю з подальшим психологічним та емоціональним аналізом;

Застосунок допоможе не тільки пройти інтерв'ю, а й переглянути результати по ньому і власні відповіді. Вбудований функціонал перевірки на справність камери та мікрофону, допоможе візуально зрозуміти чи є якісь перебої. Для цього можна включити камеру, подивитися приблизно який буде вигляд запису на самому інтерв'ю, протестувати рівень гучності голосу, за шкалою в ДБ. Проходження інтерв'ю зроблене максимально зручно та без будь-якого ускладнення задля кращого user-experience. Аналіз відповідей відбувається за допомогою окремої бібліотеки, а зіставлення тексту відповіді з правильною за допомогою алгоритму: «Левенштейнова відстань».

Даний застосунок можна використовувати не тільки в ролі допомоги при проходженні інтерв'ю, а й для внутрішнього тестування співробітників компаній або тестування осіб за бажанням покращити навички в тій чи іншій сфері.

3.3 Вимоги до застосунку

3.3.1 Загальні вимоги до застосунку

Застосунок «EasyEmployAssistance» повинен надати можливість користувачу пройти автоматичне інтерв'ю, а саме: вибрати правильний тест по заданій компетентності; перевірити наявність та справність пристроїв, необхідних для проходження інтерв'ю(камера, мікрофон); пройти тест, в якому відповіді на питання будуть розпізнані з голосу; проаналізувати відповіді, зіставити їх з правильними, і зобразити користувачу; переглянути записане інтерв'ю(для адміністраторів) та зробити психологічний або емоційний аналіз.

Кожен користувач повинен пройти автентифікацію та авторизацію для підтвердження особи та отримання відповідних прав доступу до ресурсу перед тим, як почати використовувати застосунок.

Користувач може вибрати відповідний тест, попередньо відфільтрувавши всі тести по категорії, мові програмування та рівню знань. Переглянути основну інформації по відповідному тесту: рівень складності, приблизний час проходження, кількість питань в тесті.\

Адміністратор може переглянути пройдені тести користувачів, а також відео проходження інтерв'ю для того, щоб проаналізувати користувача.

3.3.2 Вимоги до функцій, які виконуються програмою

Перелік основних функцій користувача наведено в табл. 2.

Таблиця 2. Перелік функцій користувача, що підлягають автоматизації

Функція	Задача
Реєстрація	Введення логіну та паролю як інформації про себе.
	Забезпечення вимогам по паролю (основні критерії).
	Підтвердження пошти через повідомлення з посиланням.
	Вихід з системи.
Вибір тесту для проходження	Вибір категорії тестів.
	Вибір рівня тестів.
	Вибір самого тесту після фільтрації.
Перевірка обладнання користувача	Перевірка камери.
	Перевірка звуку та зображення його рівня.
Проходження самого тесту	Автоматичне включення запису відео.
	Відповідь на запитання голосом.
	Аналіз голосу та зображення текстового варіанту в поле.
Перегляд результатів тесту	Аналіз відповідей до правильних та зображення результату в відсотках.
	Перегляд правильних відповідей та власних перетворених з голосу.

3.3.3 Технічні вимоги до застосунку

Потрібно створити вебзастосунок, що повинен містити таку функціональність:

- На основі сутностей предметної області створити модулі. Кожен модуль повинен мати одну або більше компонент.
- Кожна компонента складається з 3 файлів: TS файл, в якому лежить вся логіка; SCSS файл в якому лежать стилі; та HTML файл в якому лежить розмітка компоненти.
- В кожному TS файлі методи повинні відображати їх назви по функціям, які вони виконують.
- Сервіси повинні виконувати всю логіку взаємодії з зовнішніми сервісами та запити до API з метою інтеграції. Також громіздке обчислення повинно бути переміщене до сервісів. Так як сервіси в ангулярі наслідують патерн «Singleton», сервіси можна використовувати як місця для збереження даних протягом всього часу роботи застосунку до перезавантаження сторінки.
- Frontend частина повинна бути респонсивна та підлаштовуватися під різні розширення екрану.
- Код повинен бути документований в тих місцях, де це потрібно і не можна явно зрозуміти логіку.
- У застосунку повинна бути реалізована система авторизації та автентифікації.
- Вебсайт повинен правильно відображати контент в різних інтернет браузерях MS Internet Explorer 5.5+, Mozilla 1.7+, Opera 7.54+.
- На будь-які неправильні дії користувача, пов'язані з заповненням обов'язкових полів введення, які можуть бути оброблені програмою, генеруються відповідні блокуючі дії.

РОЗДІЛ 4.

РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

4.1 Розробка вебзастосунку

Розробка вебзастосунку «EasyEmployAssistance» осягнула декілька етапів. В наступних розділах буде сформульовано та описано кожен з них.

Провівши аналіз об'єкту та предметної області, було зрозуміло, що потреба автоматизації проходження інтерв'ю є досить актуальною в наш час.

4.2 Постановка завдання

Було виділено наступні основні завдання, які висвітлено після аналізу предметної області:

- Автентифікація та авторизація користувача;
- Надати користувачу можливість вибору категорії, складності тесту;
- Перевірка обладнання користувача: камера та звук, зображення його рівня;
- Автоматичне включення запису відео, відповідь на запитання голосом, його аналіз та зображення текстового варіанту в поле;
- Аналіз відповідей до правильних та зображення результату в відсотках, перегляд правильних відповідей та власних перетворених з голосу;
- Для адміністратора можливість переглянути відео проходження інтерв'ю, з подальшою можливістю його психологічного та емоціонального аналізу.

Після узагальнення, можна зробити висновок, що застосунок буде мати зручний інтерфейс, доступний та зрозумілий функціонал.

4.3 Структура програми

Програма, що реалізує цільовий вебзастосунок складається з клієнтської частини та third-party сервісу для автентифікації та авторизації.

4.3.1 Клієнтська частина

Клієнтська частина розроблена за допомогою фреймворку Ангуляр, який є доволі популярним в сьогодення. Для того, щоб гарно стилізувати цю частину, було використано бібліотеку Bootstrap та ангулярівський Angular Material. Структура програми складається з модулів, які в свою чергу, на основі предметної області, діляться на компоненти.

Кожен компонент має свою логіку, яка знаходиться в TS файлі. Стилі записуються в SCSS файл, який допомагає більше функціонально їх розписати. HTML розмітка забезпечує зображення даних та відповідних компонентів.

Сервіси виконують всю логіку взаємодії з зовнішніми сервісами та запити до API з метою інтеграції. Також громіздке обчислення повинно бути переміщене до сервісів. Так як сервіси в ангулярі наслідують патерн «Singleton», сервіси використовуються як місця для збереження даних протягом всього часу роботи застосунку до перезавантаження сторінки.

Кожна компонента закріплена за своїм роутом. Тобто коли ми переходимо на іншу сторінку, то змінюється посилання в адресному полі, це

зроблено для того, щоб програма була легко масштабованою та гнучкою. На кожен роут навішений Guard, який призначений для того, щоб не автентифікований користувач не зміг перейти за посиланням і отримати доступ до ресурсу.

4.3.2 Інтеграція зі стороннім сервісом

Програмний інтерфейс застосунку (Application programming interface, API) – набір готових компонент, що забезпечує доступ до відповідних функціональних частин, які надає програма та дозволяє абстрагуватися від конкретної реалізації цього застосунку.

API повинен надавати необхідні функції та відповідні параметри, задавати поведінку, а також виступати в ролі проміжної ланки для зв'язку між іншими компонентами.

Одна з ключових ідей використання цього сервісу – надати користувачу безпечний, швидкий та зручний вхід в застосунок за допомогою сервісу Auth0. Крім того, цей сервіс забезпечує зручний та простий в використанні сервіс, де можна побачити всіх користувачів сайту і ще багато чого цікавого.

Auth0 надає такий функціонал:

- Отримання деякої базової публічної інформації по користувачам, подивитися останній вхід в систему, можна додати до них ролі;
- Користувачі можуть мати можливість входити або за допомогою імені користувача/пароля, або за допомогою своїх облікових записів у соціальних мережах (наприклад, Facebook або Twitter).
- Підтвердження акаунту по SMS, коли виникли якісь підозрілі дії задля уникнення взлому.

- Проведення всіх заходів для незвичних операцій автентифікації для того, щоб уникнути DDoS-атак.
- Багатофакторна автентифікація (MFA), коли користувачі хочуть отримати доступ до конфіденційних даних.

Інтеграція відбувалася за допомогою публічного NuGet пакету, який призначений для фрейворку Angular. Модуль заімпортований в основний модуль застосунку, і після цього за допомогою DI ми можемо ініціалізувати цей сервіс і викликати методи в ньому. Після того як викликані відповідні методи для автентифікації, можна отримати інформацію по користувачу, якщо вона є (ім'я, фото, нікнейм, пошта) і дозволити йому відвідувати ресурси застосунку.

4.4 Тестування сайту

В даному розділі проводиться тестування вебзастосунку, після того, як його розробка завершена та проведено аналіз кінцевих результатів цього тестування. Ця частина роботи дає можливість зрозуміти, чи правильно виконана розробка вебзастосунку, чи весь функціонал відповідає вимогам та правильно взаємодіється.

Загалом тестування вебзастосунків відбувається в декілька етапів :

- Залучаються тестувальники для перевірки на зручність;
- Перевіряються всі переходи, посилання, кнопки і тд.
- Перевіряється можливість роботи вебзастосунку при великих навантаженнях;

Крім цього вебзастосунки перевіряють на сумісність з різними браузерами. Для цього використовується декілька популярних браузерів, запускають один і той же сценарій, потім порівнюють результати і за цими

результатами роблять висновки щодо покращень або змін для відповідного браузера, якщо знаходяться якісь невідповідності від сценарію.

В даному випадку тестування проводилось на трьох основних браузерах:

- Safari
- Google Chrome
- Microsoft Edge

4.4.1 Модульне тестування

Якість вебзастосунку — характеристика, яку намагаються тримати на високому рівні під час розробки додатку. Для цього, використовують модульне тестування, яке зменшує можливість появи дефектів.

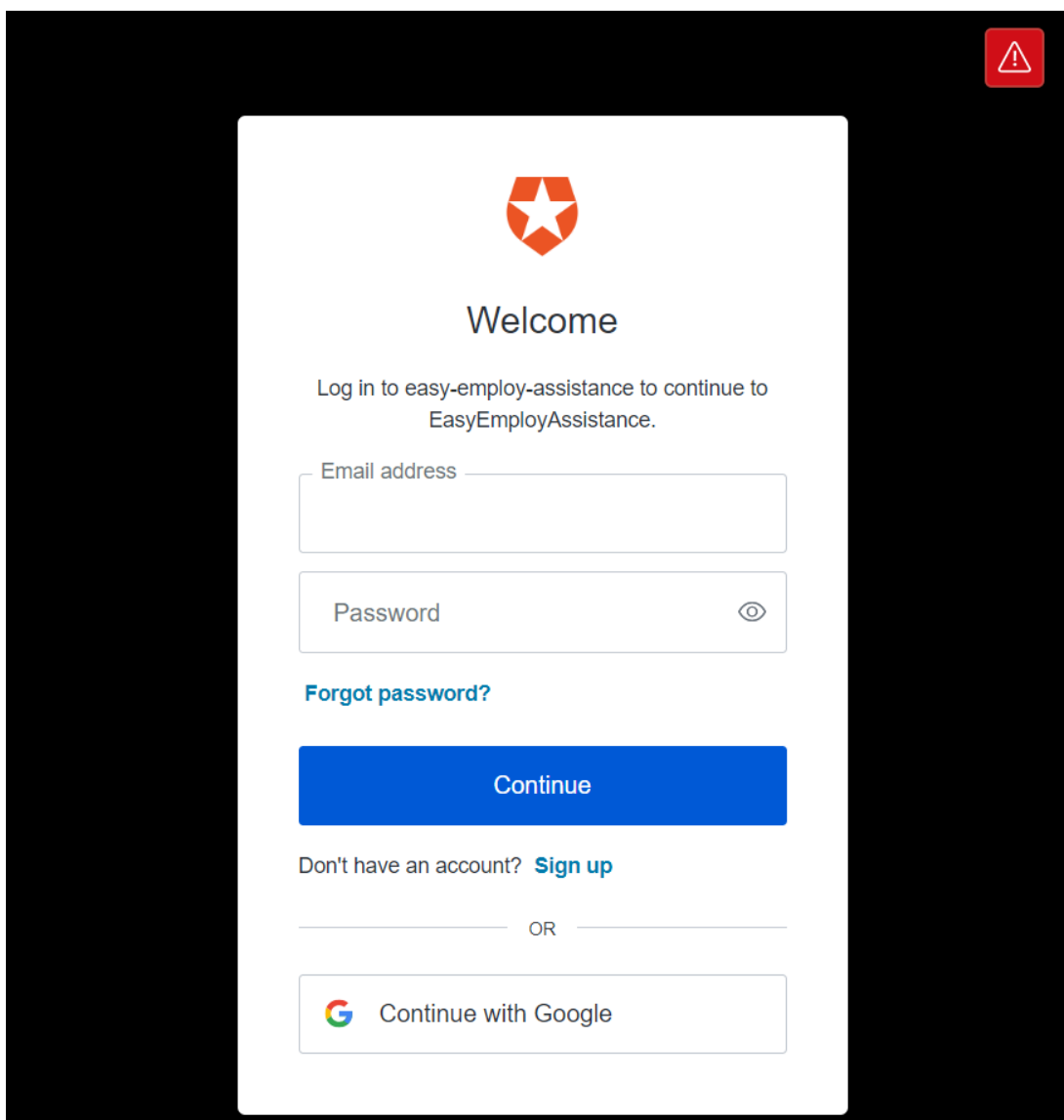
Модульне тестування — це розробка фрагменту коду, який покриває якусь відповідну функціональну зону тестованої одиниці застосунку. Частіше модульне тестування виконується за допомогою методів в певній робочій області.

При розробці застосунку використано Ангуляр бібліотеку для тестування, оскільки необхідність в цьому є зрозумілою зважаючи на вище описану характеристику. Можна дуже точно та легко протестувати окремі елементи сервісу або компоненти. Для кожного TS файлу компоненти є окремий файл для тестів цієї компоненти. Також є інші фреймворки для тестування типу ng-mock.

РОЗДІЛ 5

ІНСТРУКЦІЯ КОРИСТУВАЧА

При переході за посиланням сайту, якщо користувач не автентифікований – його зустрічає сторінка логіну або реєстрації (див. рис. 6). Для всього сайту вибрана одна мова – англійська.

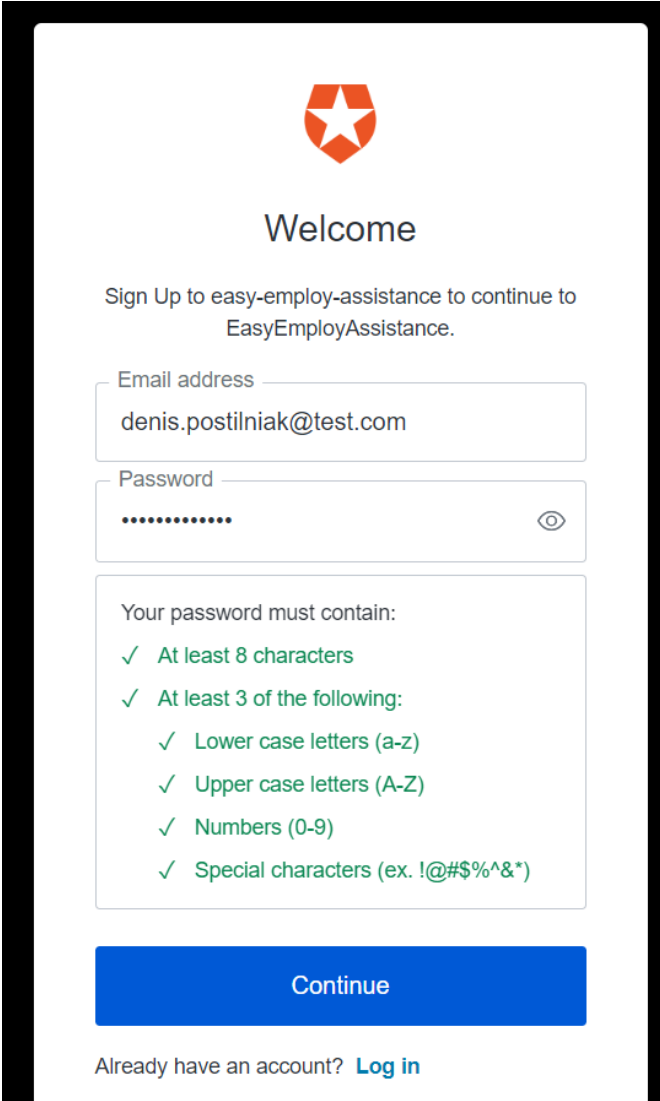



The image shows a login page for 'EasyEmployAssistance'. At the top right, there is a red warning icon. The main content is centered on a white background. It starts with a red star logo inside a shield shape. Below the logo, the word 'Welcome' is written in a large, dark font. Underneath, there is a line of text: 'Log in to easy-employ-assistance to continue to EasyEmployAssistance.' This is followed by two input fields: 'Email address' and 'Password'. The 'Password' field has an eye icon to toggle visibility. Below the password field is a blue link that says 'Forgot password?'. A large blue button with the text 'Continue' is positioned below the links. Underneath the button, there is a link that says 'Don't have an account? Sign up'. Below this is a horizontal line with the word 'OR' in the center. At the bottom, there is a button with the Google logo and the text 'Continue with Google'.

Рисунок 6. Сторінка для логіну користувача

При реєстрації користувача можна вибрати функцію SSO та зареєструватися/залогінитися за допомогою Google акаунту, або потрібно ввести пошту та придумати пароль. Пароль повинен відповідати всім критеріям, а саме (див. рис. 7):

- Мати довжину щонайменше 8 символів;
- Одна маленька та велика літери;
- Хоча б одна цифра;
- Спеціальний знак (!@#\$\$%^&* і тд.);






Welcome

Sign Up to easy-employ-assistance to continue to EasyEmployAssistance.

Email address

Password

Your password must contain:

- ✓ At least 8 characters
- ✓ At least 3 of the following:
 - ✓ Lower case letters (a-z)
 - ✓ Upper case letters (A-Z)
 - ✓ Numbers (0-9)
 - ✓ Special characters (ex. !@#\$\$%^&*)

Continue

Already have an account? [Log in](#)

Рисунок 7. Сторінка для реєстрації користувача

Далі згода на обробку персональних даних (див. рис. 8), а саме пошти та особистого профілю з базовою інформацією(це стосується як для реєстрації через SSO, так і для реєстрації через пошту і пароль).

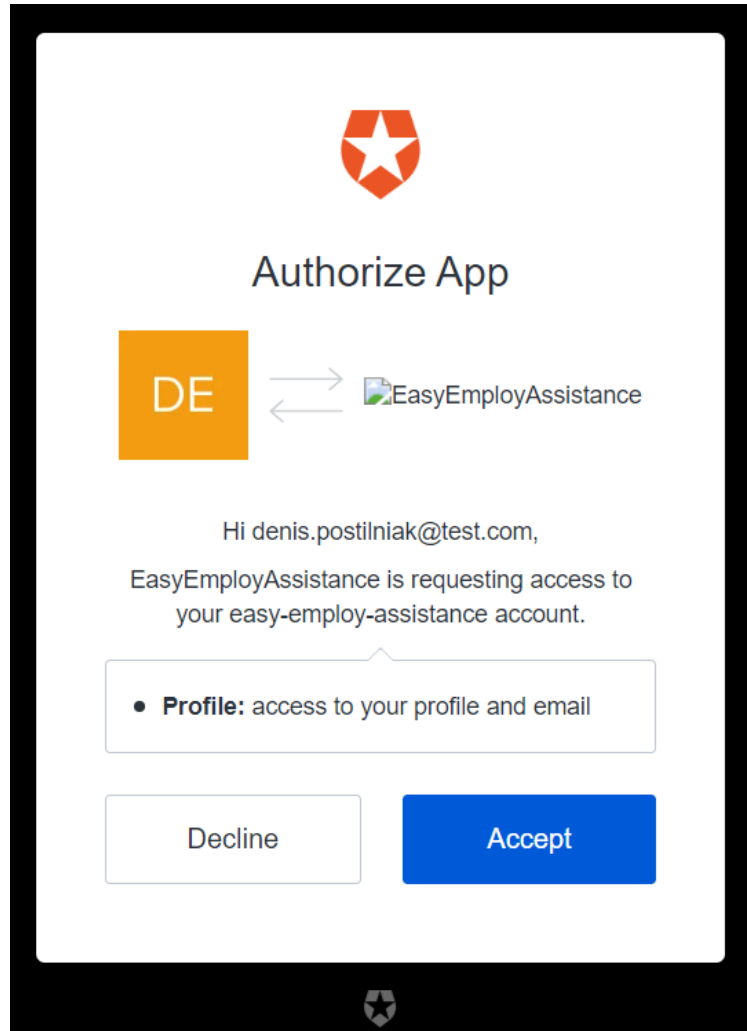


Рисунок 8. Підтвердження обробки особистих даних

Після вдалої автентифікації користувача буде перенаправлено на головну сторінку де він може вибрати тест. Користувач може вийти зі свого профілю кліком однієї кнопки. Також можна переглянути інформацію по профілю, а саме: ім'я та прізвище за наявності, та фото профілю, яке витягнуте з профілю Google якщо це SSO вхід.

Також це сторінка початку проходження інтерв'ю. Користувачу надається можливість вибрати категорію та рівень тесту. Далі всі тести

фільтруються по заданим параметрам та користувач вибирає один зі списку. По вибраному тесту виводиться основна інформація (див. рис. 9):

- Кількість запитань в тесті;
- Орієнтовний проміжок часу, який тест займе на проходження;
- Рівень складності тесту(Entry, Beginer, Medium, Hard);

The screenshot shows the 'Easy Employ Assistance' application interface. On the left is a dark blue sidebar with a user profile for 'Denis Postilnyak', 'Tasks', and 'Help' options. The main content area is titled 'Select test:' and contains three dropdown menus: 'Category', 'Level', and 'Test'. To the right, under the heading 'Information', there are three data boxes: 'NUMBER OF QUESTIONS' (Total number of questions test consist) with a value of 0, 'ETA' (Estimated time of arrival(min)) with a value of 0, and 'LEVEL' (Complexity level based on test) with a value of 'None'. At the bottom of the main area are two buttons: 'CONTINUE' and 'CANCEL'.

Рисунок 9. Інформація по профілю та вибір тесту

Наступним не менш важливим етапом є перевірка обладнання. Користувач може перевірити на справність камеру, включивши її та побачивши себе, або ж перевірки якість звуку, включивши мікрофон та перевібивши гучність за допомогою індикатору(індикатор показує рівень звуку від 0 до 1, та за кольорами: жовтий означає що погано чути користувача, а зелений показує що це нормальний рівень). Якщо все влаштовує, то можна натискати кнопку «Start» і починати тест (див. рис. 10).

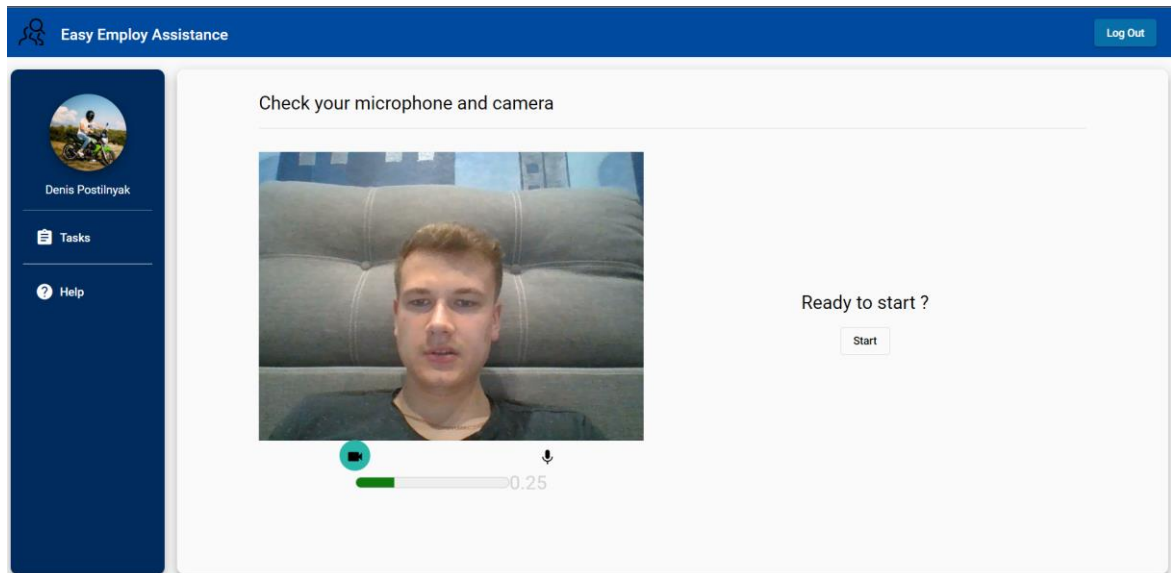


Рисунок 10. Перевірка обладнання користувача

Далі користувача застосунок перенаправляє на саму сторінку проходження інтерв'ю. З самого початку в правому верхньому кутку з'являється зображення з камери та починається відеозапис. Наступне, що ми можемо побачити – це секундомір, який показує скільки часу ми вже проходимо цей тест. Наступний елемент – це прогрес бар, який показує прогрес по питанням тесту (скільки пройдено та скільки ще залишилось).

Далі йде інформація про сам тест: це номер запитання та саме запитання в текстовому вигляді. Для того, щоб записати відповідь потрібно натиснути на кнопку запису голосу та сказати відповідь (див. рис. 11). Відповідь автоматично конвертується в текстовий варіант та зображується користувачу в текстовому полі. Якщо відповідь не чітко розпізнана або ви захотіли змінити її – можна перезаписати. І так далі аж до останнього запитання (див. рис. 12).

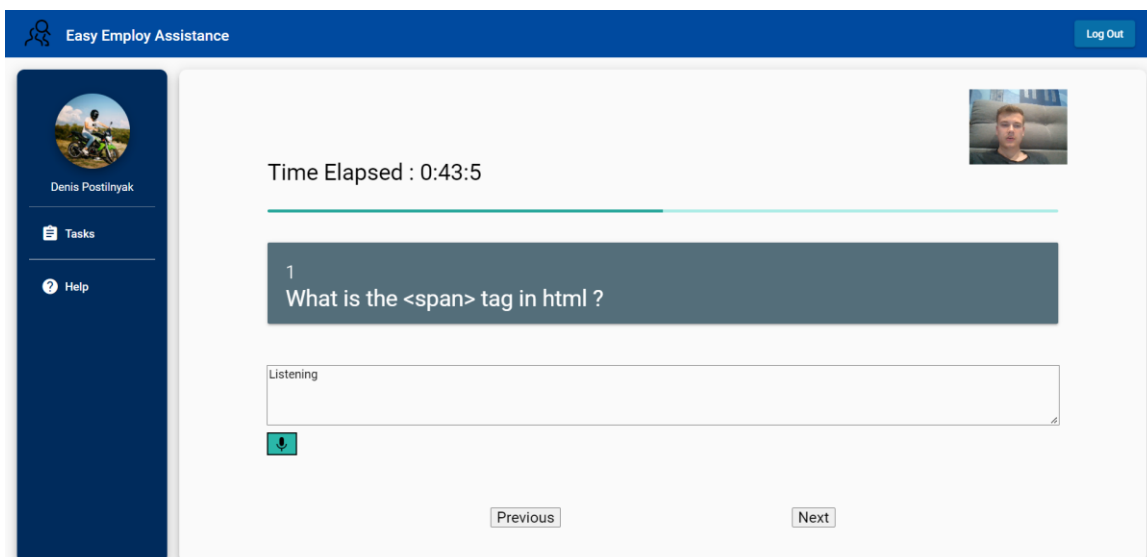


Рисунок 11. Процес запису відповіді з голосу

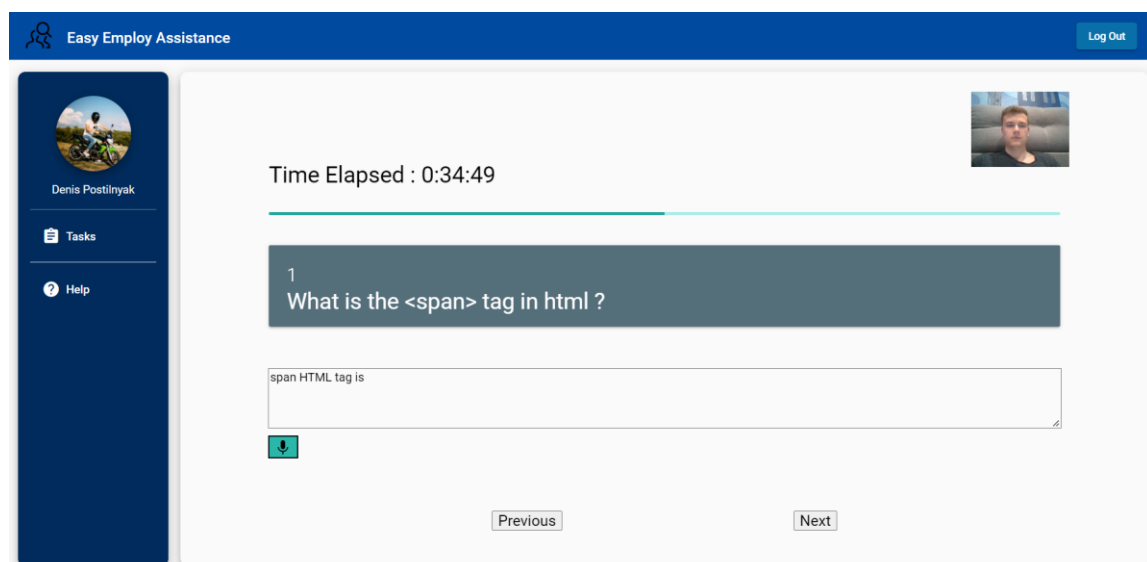


Рисунок 12. Процес проходження тесту, відповідь записана

Після того, як користувач відповів на всі питання, та розпізнані відповіді його задовільнили, користувач переходить до останнього етапу проходження – перевірка результатів. Запис відео відразу закінчується коли ми переходимо на сторінку результатів (див. рис. 13). На цій сторінці ми можемо побачити основні результати:

- Кількість пройдених тестів;
- Час проходження інтерв'ю;
- Середній відсоток правильних відповідей;
- Кожне питання з відповідями користувача та правильними.
- Записане інтерв'ю;

Користувач може побачити власні відповіді по кожному питанню тесту та правильні, за допомогою яких, додаток аналізує та зображує відсоток збіжності відповіді. Якщо відповідь набирає більше 70% збіжність – значить вона зараховується. Тест можливо повторити або перейти на сторінку вибору та фільтрації тестів.

Для більш повної картини кандидата, адміністратор може переглянути записане відео проходження інтерв'ю. Це відео можна використати як якісний, психологічний та емоціональний аналіз кандидата під час проходження тесту. Також можна уникнути деяких шахрайських дій.

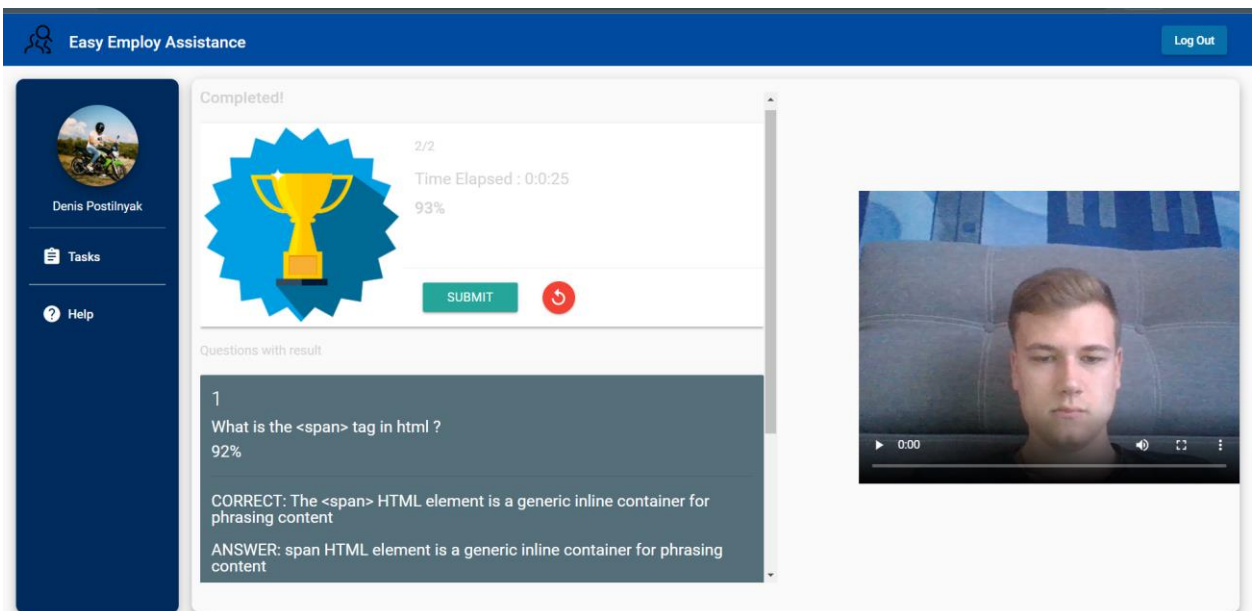


Рисунок 13. Результати тесту та перегляд інтерв'ю

ВИСНОВКИ

В роботі розглянуто засоби розробки вебзастосунків, їх основні вимоги та принципи. Проведено аналіз використання різноманітних технологій для проєктування, реалізації та тестування застосунку, інтеграції з зовнішніми ресурсами.

В роботі досліджено існуючі застосунки проходження автоматичного інтерв'ю та проведено їх порівняння. Багато функціоналу вже реалізовано, але ми вибрали найкраще та додали те, що на нашу думку покращить застосунок і додасть йому більше можливостей для розвитку.

Розроблено вебзастосунок «EasyEmployAssistance» для проходження інтерв'ю. Під час розробки вебзастосунку було виконано поставлені задачі та цілі з дотриманням вимог, а саме: проведено порівняння існуючих застосунків, розроблено застосунок для проходження автоматичного інтерв'ю, з подальшим аналізом мовлення кандидата та перетворенням його в текст; зіставлення відповідей кандидата до правильних відповідей з їх зображенням; ручним психологічним та емоційним аналізом записаного відео-інтерв'ю.

Апробовано технології (Annyang, Levenshtein dist.) для проєктування, реалізації веб-сайту та інтеграції з зовнішніми ресурсами (Auth0, Ngx-bootstrap). Для фронтенд частини було створено SPA на базі фреймворку Angular, для відображення контенту було використано інтеграцію з бібліотекою Bootstrap, логіка написана за допомогою TS, для аналізу Speech-to-text використано JS бібліотеку «annyang». Для кінцевого результату, який обчислюється за рахунок проаналізованих та правильних відповідей було використано алгоритм «Левенштейнова відстань». Також для зручнішої та більш безпечної автентифікації інтегровано застосунок з third-party сервісом «Auth0».

В подальшому слід зосередити увагу на розширенні та покращенні функціоналу даного вебзастосунку. Потенціально є дуже багато шляхів розвитку, які можуть бути реалізовані. Також дана розробка може бути інтегрована з іншими сервісами для підтримки більш розширеного функціоналу та залученням більшої кількості користувачів.

Програмний код застосунку розташовується на моєму власному акаунті в Github репозиторії і має відкритий код тому можна легко пропонувати покращення, зміни, або фікси багів якщо ті знайдено сторонніми розробниками.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Angular [Електронний ресурс] – Режим доступу до ресурсу: <https://angular.io/>.
2. Auth0 [Електронний ресурс] – Режим доступу до ресурсу: <https://auth0.com/>.
3. Annyang [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/TalAter/annyang>.
4. Yakov F., Moiseev A. Angular Development with TypeScript: Manning, 2008. 560 с.
5. Modern Hire [Електронний ресурс] – Режим доступу до ресурсу: <https://modernhire.com>.
6. Cammio [Електронний ресурс] – Режим доступу до ресурсу: <https://cammio.com/>.
7. Indivizo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.indivizo.com/>
8. Levenshtein distance [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Levenshtein_distance
9. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>
10. Ng-mock [Електронний ресурс] – Режим доступу до ресурсу: <https://ng-mocks.sudo.eu/>
11. Design patterns to know in Angular [Електронний ресурс] – Режим доступу до ресурсу: <https://dev-academy.com/angular-architecture-best-practices/>
12. A Complete Guide to Flexbox [Електронний ресурс] – Режим доступу до ресурсу: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
13. TypeScript [Електронний ресурс] – Режим доступу до ресурсу: <https://www.typescriptlang.org/>