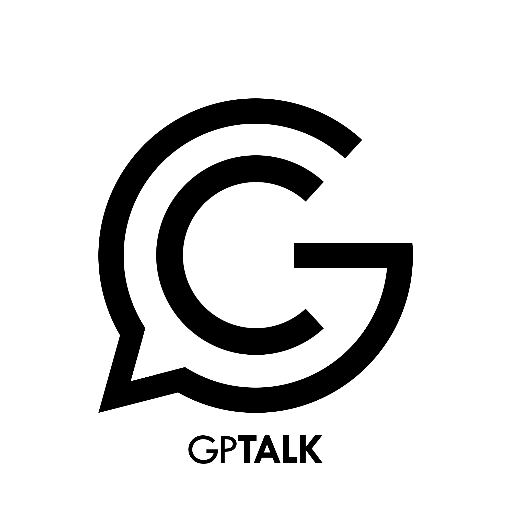


Capstone Project Phase A

Software Engineering Department

Braude College

Project number: 24-1-D-11



GPTalk – Learning languages, enhanced for the modern age

**Supervisor**  
Alex Keselman

**Students**  
Oneill Panker [oneill.panker@e.braude.ac.il](mailto:oneill.panker@e.braude.ac.il)  
 Roman Gury [roman.gury@e.braude.ac.il](mailto:roman.gury@e.braude.ac.il)

Table of Contents

[Abstract 3](#_Toc166695668)

[Introduction 3](#_Toc166695669)

[Introduction to the problem 3](#_Toc166695670)

[Other solutions 4](#_Toc166695671)

[Our solution 4](#_Toc166695672)

[Who need the solution and how it will help 4](#_Toc166695673)

[Related Work 5](#_Toc166695674)

[Duolingo 5](#_Toc166695675)

[Babbel 5](#_Toc166695676)

[ChatGPT 5](#_Toc166695677)

[Disadvantages 6](#_Toc166695678)

[Web Development 6](#_Toc166695679)

[Cross-Platform software 7](#_Toc166695680)

[Angular 7](#_Toc166695681)

[NodeJS 8](#_Toc166695682)

[Express.js 8](#_Toc166695683)

[MongoDB 8](#_Toc166695684)

[Gamification 9](#_Toc166695685)

[Expected Achievements 9](#_Toc166695686)

[Engineering Process 10](#_Toc166695687)

[Process and motivation 10](#_Toc166695688)

[Constraints 12](#_Toc166695689)

[Product 13](#_Toc166695690)

[About the App 13](#_Toc166695691)

[Features 13](#_Toc166695692)

[Function & Non-Functional Requirements 14](#_Toc166695693)

[Functional: 14](#_Toc166695694)

[Non-Functional: 14](#_Toc166695695)

[Application Architecture 14](#_Toc166695696)

[Mockups 15](#_Toc166695697)

[Diagrams 20](#_Toc166695698)

[Activity Diagrams 20](#_Toc166695699)

[Package Diagram 21](#_Toc166695700)

[Deployment Diagram 22](#_Toc166695701)

[Verification Plan 22](#_Toc166695702)

[References 23](#_Toc166695703)

# Abstract

More people than ever are learning foreign languages. Our project, GPTalk, is a cutting-edge language learning application designed to enhance the user's language acquisition experience by utilizing the latest advancements in artificial intelligence. The app takes advantage of the capabilities of the ChatGPT API to generate personalized, fresh exercises and quizzes, addressing issues such as variety and informative feedback that are often found in other language learning platforms. With an interactive and user-focused approach, GPTalk offers infinite possibilities for learning material and explanations of mistakes, ensuring a superior user experience. Additionally, the app provides virtual conversation practice with the ChatGPT bot, allowing students to hone their skills in a supportive environment.   
By providing adaptive learning paths for beginners and challenging experiences for advanced learners, GPTalk positions itself as a comprehensive solution in the world of language learning, offering a unique and engaging approach to help users succeed in their language acquisition journey.

# Introduction

## Introduction to the problem

Learning a second language has become standard in the modern age. An estimated 50% of the world population is at least bilingual (Grosjean, 2019) and the acquisition of new languages is standard both in primary and secondary education across the globe. There are numerous reasons for the need to learn new languages, from cultural enrichment to financial benefit, or the result of immigration to a foreign country, and this demand is ever present.  
While countless students take language courses every day, far fewer students see them to completion: studies on the retention rate of students taking online courses show exceptionally low retention rates both in formal online courses (Angeles Sánchez-Elvira Paniagua, 2018) and in open non-formal ones (Neil Cowie, 2019), indicating that there is room for improvement when it comes to online learning methods. Among the reasons for high student dropout rates are the need for self-motivation to follow through with the courses and a lack of structure. There is also a need for clear instructions and examples that online courses do not always provide and are a cause for confusion and frustration for the students. Many students taking online language courses also tend to be older than conventional students and must balance the learning process with their families and work. Furthermore, online courses often lack proper support for the students: A question cannot always be answered in real time. (Angeles Sánchez-Elvira Paniagua, 2018)

## Other solutions

In the world of language learning, there have been significant advancements with the integration of digital technologies. Popular apps like “Duolingo” and “Babbel” have introduced varied methods such as gamification and immersive learning to make language acquisition more engaging. Duolingo, the most popular of these apps, focuses on teaching through gamification, a strategy which integrates gaming concepts into the learning process.  
Studies about the user experience in the app show that participants found the activities to be repetitive (Shawn Loewen, 2019) and limited in terms of grammatical explanations. (Marques-Schafer, 2018)  
The advancement of adaptive technologies like ChatGPT API is a step forward in offering personalized, AI-enhanced learning, allowing for tailor-made learning experiences. With high scores, levels and more gamification elements, the learning experience becomes more interactive and enjoyable.  
Gamification will make learners more motivated to continue the language learning journey. These developments represent a shift toward a flexible, user-focused learning process, supporting distinctive styles and needs. Despite these improvements, there’s still room for a more comprehensive, engaging language learning tool.

## Our solution

GPTalk aims to provide a superior learning experience by utilizing the latest developments in artificial intelligence to generate the learning material. The utilization of the ChatGPT API will help address issues both in terms of variety and informative feedback: Every time the user accesses the app to improve in the language of his choice, he will receive fresh questions and exercises, generated from the GPT-4 language model’s vast database. The language model’s advanced features will also have no problem providing the user with explanations of his mistakes and generating further examples of relevant concepts if needed. It will also be possible to practice what we have learned up to that point by having a conversation with the ChatGPT bot, providing the student with a virtual teacher that will help him hone his skills.  
All this guarantees infinite possibilities in terms of exercises, quizzes, and general information, making for a superior user experience compared to what is available today on the market.   
Of course, learning a foreign tongue exclusively through a web app is not a substitute for real life practice by having conversations with native speakers, or exposure to various forms of media in the relevant language. However, a well-designed application that utilizes the latest developments in artificial intelligence can help immensely in putting beginners on the right track and give them the initial tools to succeed in their journey of acquiring a new language.

## Who need the solution and how it will help

The target audience for this type of website starts from casual learners that seek to learn for personal enrichment and for students and educators who seek fluency for academic and career reasons.  
Considering that the website will use AI-enhanced learning for more tailor-made learning, the website will help any level of learner. Beginners will get an adaptive learning path to build their proficiency in a gradual manner, and the more advanced learners will enjoy challenging experiences to enhance their confidence.

# Related Work

DuolingoDuolingo is a popular educational app, which utilizes a gamified approach to teach its users different languages. The app focuses on short “bite-sized” lessons that involve various interactive exercises and quizzes in order to make the learning experience more fun and engaging for the user. Students receive experience points for correct answers, various daily goals are given for the users, and completing lessons on a daily basis is counted towards a streak to incentivize continuous use of the app. Like many videogames, Duolingo also has a “heart” system, where incorrect answers cost the user hearts. When the number of hearts is reduced to zero, relevant lessons and drills are provided to help the student learn from his mistakes. (Shortt, 2023)  
The app also uses an algorithm that adapts the difficulty level of the exercises to each user and provides unique feedback and recommendations.  
Along with teaching commonly used phrases, Duolingo also uses a strategy of exposing the learner to more unusual sentences, that can be more memorable due to their quirkiness, and can push the user to carefully think about the language they are learning. (Duolingo, 2024)

Duolingo is available as a free app in its basic form and offers a subscription model called “Super Duolingo” that includes numerous benefits such as extra hearts and removes ads.  
Currently, Duolingo is the most popular language-learning app in the world, with over 500 million users. (Duolingo, 2024)

## Babbel

Babbel is a language learning application that offers a conversational approach to learning languages. The learning is tailored to fit the user’s native language.  
Babbel uses a practical methodology that creates real-life contexts through short and interactive lessons.  
The app has several levels of proficiency, from beginner to advanced.  
Babbel focuses more on structured lessons and less on gamification, but it includes several elements like progress tracking and achievements to motivate learners.

## ChatGPT

ChatGPT is an artificial intelligence content generating model, presented as a chat application where the user talks to a bot, which provides detailed and accurate responses according to the user’s prompts. The model can perform a wide array of tasks, from translation to summarization, the generation of ideas, stories, poems, scripts, and letters. The model can provide coherent answers, similar to what a user would expect from a human. ChatGPT can also remember its conversation history with the user, allowing it to partake in continuous dialogue.

In order to train its abilities, ChatGPT uses a mix of supervised learning and reinforcement learning based on human feedback.

* Supervised learning involves using labeled datasets to teach agents the relationship between inputs and outputs and consequently train them to predict outcomes and recognize patterns.
* Reinforcement learning allows the agent to learn through trial and error, giving it a "reward" - feedback for the decisions it makes, either positive or negative depending on the goal. Unlike supervised learning, a correct set of actions does not exist in reinforcement learning: The agent finds the optimal set of actions that gives it the maximum cumulative reward.

To train ChatGPT, humans teach the model by engaging in conversations simulating real-life usage of the application: Both the queries and the responses are provided by the human, and this information is used as the training data for the model. (OpenAI, Introducing ChatGPT, 2022)

ChatGPT uses a large language model called GPT (Generative Pre-trained Transformer) – an artificial neural network trained on broad data, to receive training which allows it to generate proper responses to the user’s queries. The neural network contains many millions of parameters – the material that the algorithm learns through training. The most recent version, GPT-4, accepts both images and text as input and is estimated to contain around 1.7 trillion parameters. (Bastian, 2023)

Various plugins exist for ChatGPT, each one focusing on performing different tasks, such as programming, creative writing, trip planning, solving math problems, tech support and many more. (OpenAI, ChatGPT Plugins, 2023)

## Disadvantages

While Duolingo and Babbel have revolutionized the way individuals learn languages, offering accessibility and a wide range of languages, they are not without their disadvantages. Duolingo's gamified approach, though highly engaging, can sometimes emphasize repetition over deep language understanding, leading learners to excel in app-based tasks without necessarily achieving fluency or conversational competence. Babbel, with its structured lessons, offers a more traditional learning path. Yet, it may not fully cater to learners seeking to practice spontaneous speaking skills or understand the nuances of casual conversation. Additionally, both platforms have a one-size-fits-all approach to language learning, which might not suit all learning styles or meet specific educational goals, such as professional language use or specialized vocabulary acquisition.

## Web Development

Web development is the work involved in creating, building, and maintaining websites and web applications for the internet. The process includes visual design, programming, and database management.  
The basic tools involved in web development are HTML (Hypertext Markup Language) which is used to define the content and basic structure of a website, CSS (Cascading Style Sheets) which are used for presentation, styling, and JavaScript, which handles the functionality and interactivity.  
Web development is typically divided into Front-end and Back-end development: Front-end refers to the client side, the part that the user interacts with, and the Back end refers to the functionality of the web site – database interactions, user authentication, server configuration and business logic. (BrainStation, 2024)

## Cross-Platform software

Cross-platform software are programs designed to function on more than one computer architecture or operating system. Cross-platform development allows the developer to code assets only once without the need to manually port them to other platforms, saving development time and financial cost.   
Web applications are considered cross-platform, since, if they are written properly, they are accessible from any browser on any device, whether it is a computer, an iPhone, or an Android phone.   
In order to provide an optimal user experience across devices, cross-platform web apps implement Responsive web design, an approach that allows the application to be properly viewable on a variety of hardware and on different screen sizes. (Awati, 2023)

## Angular

Developed and maintained by Google, Angular is a robust framework for building mobile and web applications. Angular is a TypeScript-based framework that helps with the developer experience. With an architecture based on components, it is a great solution for creating scalable single-page applications (SPAs) (Angular.io, 2023). With this architecture, the developer can reuse components with less effort. As Angular is the second most popular front-end framework (Overflow, 2023), it has a lot of tools and a helpful community. Its integration with various back-end technologies and support for advanced features like lazy loading, server-side rendering, and dynamic content management further extend its capabilities. Additionally, Angular's powerful data-binding features streamline the process of building interactive and dynamic user interfaces, enhancing the overall user experience. The framework's emphasis on testability and performance, with tools and services like Angular Universal for server-side rendering, helps in building highly performant and SEO-friendly applications. Angular's continuous updates and improvements, guided by Google's long-term support, ensure that it remains at the forefront of modern web development technologies, making it an ideal choice for developers looking to build high-quality, scalable web applications. A key part of Angular's ecosystem is RxJS, a library for reactive programming that allows developers to handle asynchronous data calls, concurrency, and event-based programs with ease. RxJS's observables offer a powerful way to work with streams of data, which integrates seamlessly with Angular's data binding and event management infrastructure, thereby enabling more efficient and sophisticated data handling and UI updates.

## NodeJS

NodeJS is a server-side JavaScript platform used to write web applications and build servers. It is designed for high performance and scalability, and its architecture is well suited for the development of full-stack applications. Unlike many of its competitors that rely on threading to operate, NodeJS works using an event driven approach: The platform utilizes a simple event dispatch model that relies on JavaScript async functions and Promise objects. This allows NodeJS to support tens of thousands of concurrent connections without being slowed down by context switching, as is the case in platforms that rely on threading. The model makes NodeJS ideal for building applications that rely on real-time communication between the server and client.   
NodeJS allows the programmer to implement applications with the same programming language both on the client and the server side, which offers numerous benefits such as the ability to easily migrate code between the server and the client, the existence of common data formats and software tools between both sides, and the ability for the same programming staff to work on both. (Panchal, 2024) (Herron, 2020)  
These advantages led to widespread adoption of the platform around the world, with NodeJS being used in many modern web applications, from Netflix, to GitHub, PayPal, Spotify and many more. (Herron, 2020)

## Express.js

Express.js is a framework used for building web applications and APIs using Node JS. The framework simplifies building backends for web apps, allowing developers to save time and focus on the logic of the application without the need to manually write the boilerplate code (setting up route handlers, ports etc.). (Foster, 2023)  
Express.js has several features that makes building web applications easier and faster:

* **Middleware Ecosystem** – Middleware functions, which can be easily integrated into applications, increase the functionality of Express by allowing the programmer to handle various tasks, such as authentication and error handling.
* **Scalability** – The framework is designed to be lightweight and scalable, and like NodeJS itself, it is asynchronous and has an event-driven architecture, allowing it to handle a large number of requests.
* **Simplicity** – Express.js is easy to learn thanks to its straightforward design, allowing the user to build servers, set up routes and handle HTTP requests quickly.
* **Flexibility** - Express is an un-opiniated framework, allowing the developer more freedom in terms of structuring their code instead of forcing a particular code structure. (Sharma, 2023)

The framework is the most popular one for the NodeJS platform; Among the companies using Express.js are Uber, IBM, and PayPal.

## MongoDB

MongoDB is the most popular NoSQL database management system (DB-Engines, 2024). It is based on documents, which provide MongoDB with the flexibility and scalability needed for complex queries. The document model is structured like a JSON file, which makes it intuitive for developers to work with, as they can store data in a format that closely resembles their application's data. MongoDB supports indexing, replications, load balancing, and more features to make it faster and more scalable. Its architecture allows for horizontal scaling and the distribution of data across multiple servers, enabling it to handle large volumes of data and high traffic loads efficiently. This scalability feature is a key advantage for growing applications, ensuring they can expand without the need to redesign the database structure.  
Moreover, MongoDB's dynamic schema design allows for the easy modification of data and structure without downtime, facilitating agile development and rapid iteration. This flexibility is particularly beneficial in environments where requirements change frequently. The database also offers a powerful query language that supports complex searches, aggregations, and data analysis, enabling developers to retrieve and manipulate data effectively.  
Additionally, MongoDB's ecosystem includes extensive documentation, client libraries for various programming languages, and tools like MongoDB Atlas, a fully managed cloud database service. This ecosystem, combined with a supportive community and enterprise-level support, makes MongoDB a comprehensive solution for developers looking to leverage the power of NoSQL databases for their applications. Its robust feature set and ease of use have made it a go-to choice for startups and large enterprises alike, driving the adoption of MongoDB across a wide range of industries and applications.

## Gamification

Gamification in learning is an educational approach that adds video game elements inside learning environments. This approach aims to maximize the learner's enjoyment and make the learning process more interesting.  
In this era of learning apps, many of them include gamification elements.  
In addition to improving engagement in the learning process, gamification also improves retention rates.  
In this digital fast-paced world, gamification can bridge the gap between normal learning methods and video-game-like immersive experiences.

# Expected Achievements

We expect to develop a working web application that will have a positive effect on the process of language learning by making the experience more fun and less monotonous. The app will have a variety of exercises generated with the help of the ChatGPT API and support multiple languages that the user will be able to learn. The app will have various elements of gamification to encourage the user to keep going, including daily goals, streaks, and achievements to acquire.  
GPTalk's interface will be clear and readable for the user in order to help them focus on the learning process. Furthermore, the application will be able to provide clear explanations based on the learner's answers and queries and give various suggestions and bits of information that would contribute to a sense of structure for the user.

# Engineering Process

In the following section we will go over the process of writing our application, elaborate on the difficulties we faced during development, discuss the reasoning for our choices of frameworks and libraries, and explain the constraints and their influence on the development process.

## Process and motivation

The main reason for developing a language learning application was the untapped potential we saw in language models such as GPT to be used as a tool to provide learning material for the user. The enormous amount of data provided by the model, as seen in (Bastian, 2023) , if accessed using the right queries, could provide students with a near-infinite number of exercises and learning material, far surpassing any single book or learning app available on the market. We saw the ability to dynamically generate material suited to the user’s needs as a valuable tool to help the learner get acquainted with the language he is studying.

Results seen in studies such as the ones discussed in (Shawn Loewen, 2019) and (Marques-Schafer, 2018) showed us further proof that the current solutions for language learning had room for improvement, which our application attempts to achieve.

Our engineering process for developing GPTalk was planned while taking into account the unique requirements of creating a language learning platform powered by gamification and the ChatGPT API. Initially, we delved into the capabilities and functionalities of the ChatGPT API, conducting research to understand how it could be effectively utilized to generate engaging and educational language learning tasks. This foundational step was crucial in shaping the application's core learning mechanics.

Following this, our focus shifted towards the architectural and technological considerations necessary to bring GPTalk to life. We selected a tech stack that would not only ensure robust performance and scalability but also align with our vision of creating a seamless and interactive user experience.   
For the backend, we chose Express.js, a web application framework for Node.js. This choice was influenced by several factors:

* Express.js provides a powerful and flexible routing system, making it easy to define and handle routes for different HTTP methods.
* The framework is known for its performance and scalability, making it suitable for building efficient server-side applications, capable of handling the dynamic content generated by the ChatGPT API.
* Express.js supports an extensive middleware ecosystem, allowing developers to integrate various middleware functions for tasks like authentication, logging, and error handling.

Our framework of choice for the project was Angular, due to our prior experience with it, and several powerful features for building dynamic single-page applications (SPAs), which we found preferrable to the competition:

* Angular is component based, enabling us to develop reusable and maintainable code structures, and making the workflow much more organized compared to libraries like React.
* Angular is opinionated, meaning the developer must follow a consistent set of design patterns, as opposed to the competition, where much more ways of constructing the application exist, meaning that every developer will have a different method of implementing various basic features, and consequently, solving certain issues or taking over other developer’s code could become more difficult. Therefore, we see Angular’s more restrictive approach as a plus.
* Angular is built from the ground up with TypeScript in mind, a language that has several advantages over JavaScript, such as being more organized, having extra features, and being able to spot compilation errors before running the code.

For data management, MongoDB was selected as our database solution. There were several reasons for our choice:

* MongoDB’s NoSQL nature was ideal for storing the flexible and varied data types associated with language learning content, user profiles, and interaction histories.
* Its scalability and performance, coupled with its ease of integration with Node.js, made it an optimal choice for managing the application's data needs efficiently.
* As a result of MongoDB’s popularity, extensive documentation and resources are available for the platform, which meant that using the program and finding solutions to related issues would be more straightforward.

The main part of the GPTalk puzzle is of course the LLM we would choose to handle the application’s main tasks: providing the user with accurate and diverse responses, exercises, and general educational information. Without a sufficiently capable language API, none of those features would properly work, making the app unreliable. Although with current language learning model technology getting 100% accurate responses is not yet possible, OpenAI’s GPT-4 is one model that getting close to such results, and its capabilities in our project’s area of interest put it ahead of the competition:

* Unlike most other LLM’s, GPT-4 is multilingual, supporting numerous languages other than English and capable of accurate translations in many of them. (Achiam, Gpt-4 technical report, page 8, 2023)
* GPT-4 has by far the most parameters of the available LLMs (around 1.75 trillion), giving it an edge when it comes to producing accurate responses to the user’s queries. (Patricio Cerda Mardini, 2023)
* The model achieved impressive test results in relevant simulated tests such as the  
  GRE Verbal exam, where it scored at the 99th percentile, surpassing the most capable students and the previous iterations of the model. (Achiam, Gpt-4 technical report, page 6, 2023)

By integrating the aforementioned technologies and methodologies, we crafted a comprehensive and engaging language learning platform that leverages the innovative potential of the ChatGPT API, enriched by gamification elements to enhance motivation and learning outcomes.

## Constraints

In developing our project, we encountered several challenges related to our chosen technologies. The primary challenge was ensuring compatibility and seamless integration between these technologies to create a cohesive application.

**ExpressJS**, while versatile and efficient for building our backend, required a deep understanding of asynchronous programming and middleware management. Ensuring security and efficient data handling was crucial, given our reliance on real-time data processing.

**Angular** presented a steep learning curve due to its comprehensive framework and TypeScript base. The component-based architecture, although beneficial for our project's scalability, required a careful approach to component design and state management to avoid performance bottlenecks.

**MongoDB**, our choice for the database, introduced complexities in schema design and data consistency. The non-relational nature of MongoDB demanded thoughtful consideration of data structure and query optimization to ensure fast and reliable access to data.

**ChatGPT API** poses anticipated challenges, notably in ensuring the future accuracy and relevance of its responses for our language learning platform. While the API's capability to generate dynamic content holds immense potential, the prospect of inaccuracies remains a concern that could impact the educational effectiveness of our application. To proactively address this, we will be crafting precise and context-aware prompts designed to steer the API towards generating content that answers our various requirements. Some examples of the parameters given to the API are as follows:

* Specifying the output format (JSON), without additional text in the response.
* Using a prompt template to allow the use of dynamic parameters in the prompts.
* Limiting the number of questions in the response.
* Specifying language of the exercise we are generating.
* Specifying the difficulty of the generated exercise.
* Specifying the type of the generated exercise.
* Requesting the generated response to be printed in a specific structure.
* Usage of various keywords to get a more accurate result.

The anticipated refinement process will involve a systematic approach to testing and adjusting our interactions with the API. This will be crucial for identifying the optimal strategies to elicit accurate and relevant responses. A deep dive into prompt engineering will be necessary to understand how nuanced queries can influence the quality of the generated content, ensuring it meets our strict standards for educational content.

In addressing these challenges, we leaned on best practices, community support, and iterative testing. This approach allowed us to navigate the constraints effectively, ensuring our application was robust, user-friendly, and scalable.

# Product

## About the App

In the planning stages of GPTalk, our application is designed to offer a distinctive approach to language learning by incorporating elements of gamification alongside the ChatGPT API. This platform aims to enrich the language acquisition process, making it more interactive and engaging for users. The application will utilize the ChatGPT API to create personalized tasks, quizzes, and dialogues, aiming to provide a learning experience that is tailored to the individual needs and progress of each user. Gamification features such as points, levels, and achievement badges are intended to encourage learners by introducing elements of challenge and reward into the educational process.

## Features

* **Multi-language support:**  
  **The application will provide a selection of languages for the user to learn. The available languages will be the ones where GPT-4 is most familiar with, to guarantee more accurate prompts and responses. The user’s progress in each language will be tracked and saved in the database.**
* **Personalized language learning paths**:  
  To get a more tailored learning experience, the user will fill out a form where they help the app to determine their ability in the language of their choice. The information from the filled out form will allow GPTalk to generate a more personalized learning experience for each user in terms of exercise difficulty and learning subjects.
* **Interactive dialogues with ChatGPT:**  
  **The user will be able to engage in simulated dialogue with GPTalk, a feature that will allow the students to improve their verbal communication skills.**
* **Daily challenges and quizzes:**  
  **To offer a more engaging and fresh learning experience, The app will feature daily tasks that will encourage the user to consistently practice and improve their language skills.**
* **Gamified progress tracking:**  
  **The app will also motivate users by tracking their learning progress through a system of points, levels, and badges: Correct answers will make the user gain experience points that will allow them to progress towards the next level, and achieving various milestones will reward the user with completion badges.**
* **Social features:**  
  GPTalk will build **a sense of community and friendly competition among users by featuring high scores and leaderboards, encouraging users to excel and engage more deeply with the content.**

## Function & Non-Functional Requirements

### Functional:

* The application will base its learning contents on responses given by the ChatGPT API.
* The application will generate daily learning material.
* The application will allow users to register and create personal accounts.
* The application will provide secure login/logout functionality.
* The application will track and display user progress using points, levels and badges.
* The application will include milestones to encourage continued learning.
* The application will support multiple languages for learning.
* The user will be able to see the leaderboard for each language.

### Non-Functional:

* The application will be user-friendly, with an intuitive interface suitable for all ages.
* The application will feature responsive design, ensuring compatibility with desktops and various mobile devices.
* The application will provide the student with a smooth user experience and will process his inputs with minimal latency.
* The processing of each request by the API should be done within 10 seconds.

## Application Architecture

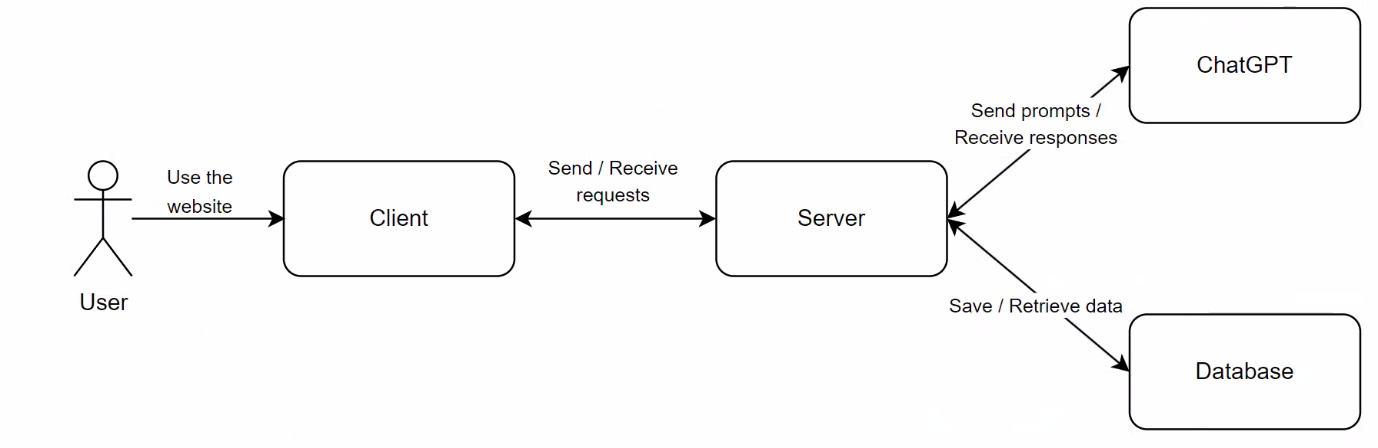


Figure 1 GPTalk architecture diagram

## Mockups



Figure 2 Launch Page featuring language selection

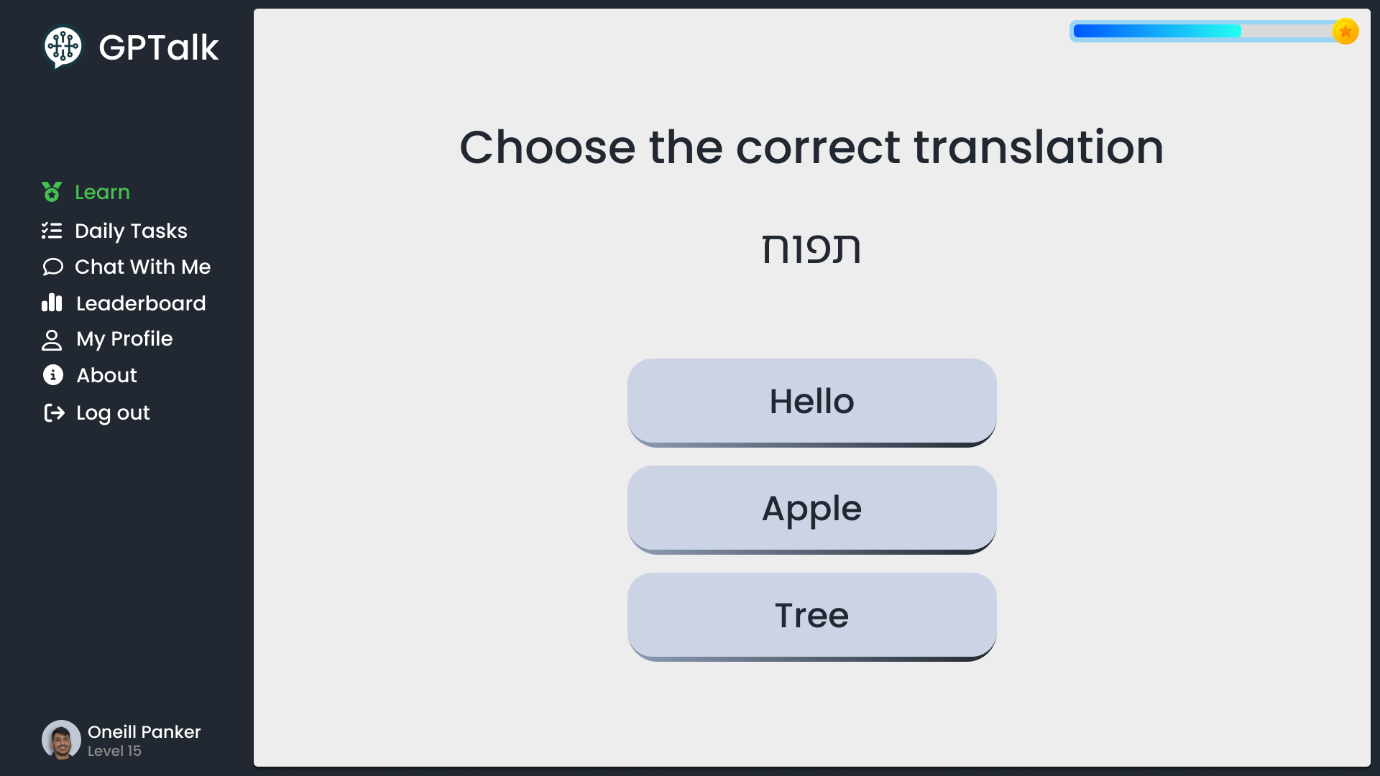


Figure 3 Exercise - variant 1

A screenshot of a computer

Description automatically generated

Figure 4 Exercise - variant 1 - correct answer

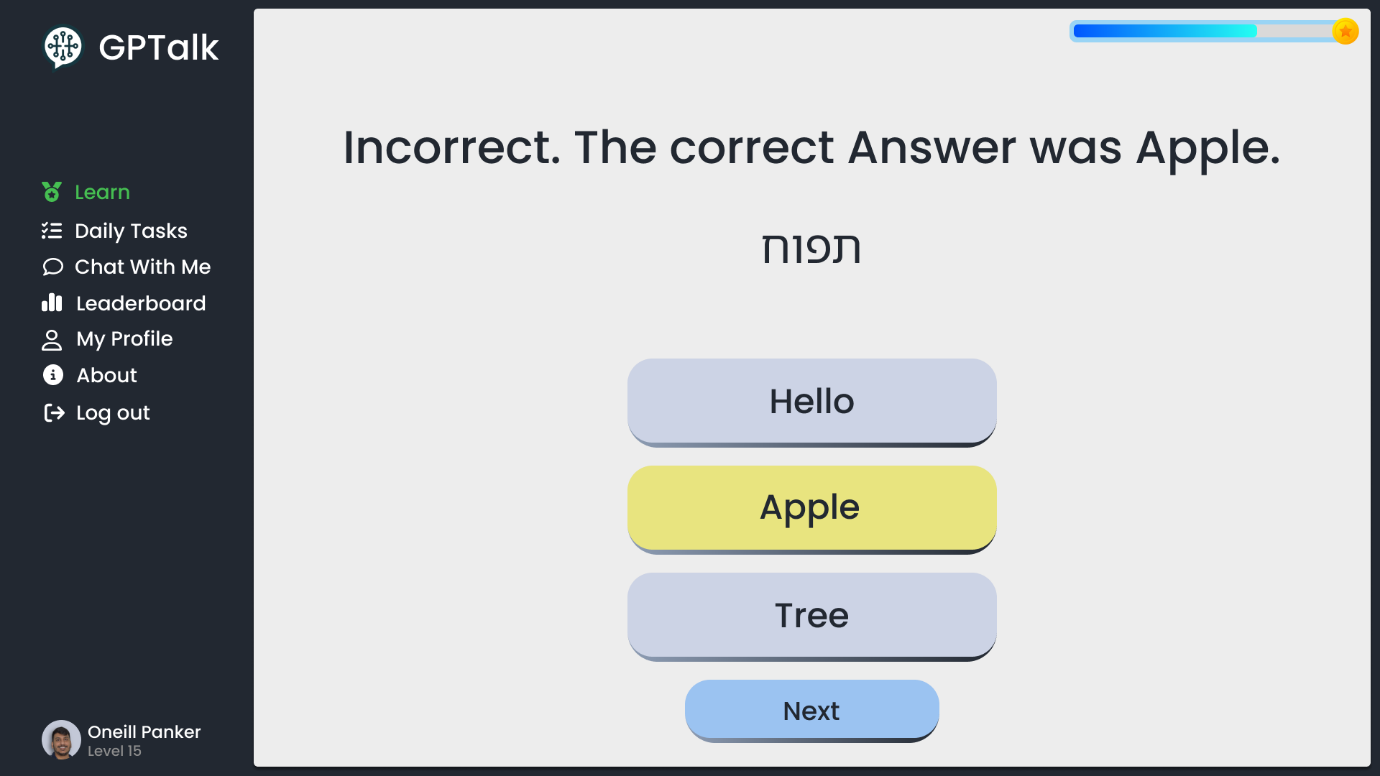


Figure 5 Exercise - variant 1 - incorrect answer

A screenshot of a computer

Description automatically generated

Figure 6 Exercise - variant 2

A screenshot of a chat

Description automatically generated

Figure 7 Exercise - variant 3

A screenshot of a computer

Description automatically generated

Figure 8 Exercise - variant 4

A screenshot of a computer game

Description automatically generated

Figure 9 Exercise - variant 5

A screenshot of a test

Description automatically generated

Figure 10 Exercise result page

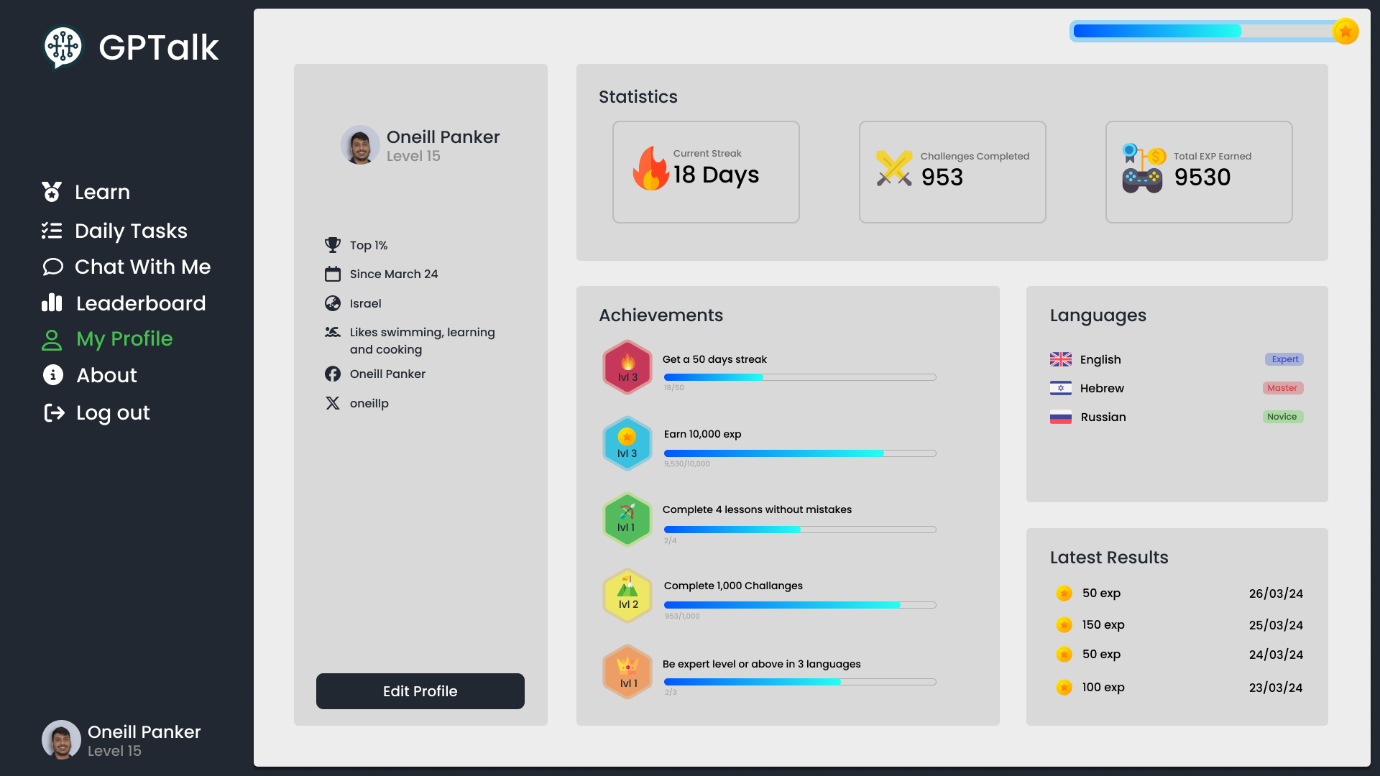


Figure 11 My Profile page

## Diagrams

### Activity Diagrams

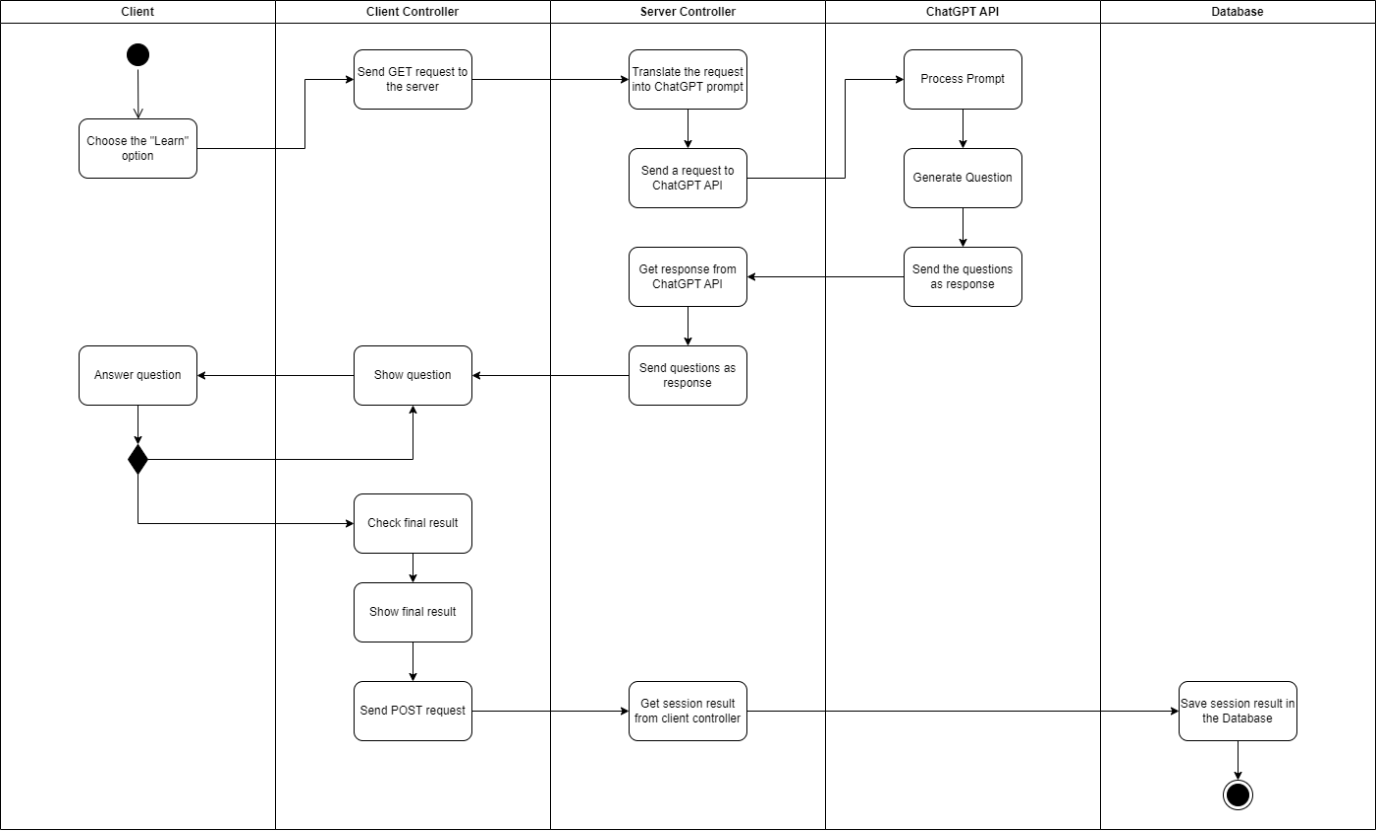


Figure 12 Activity Diagram - learning session

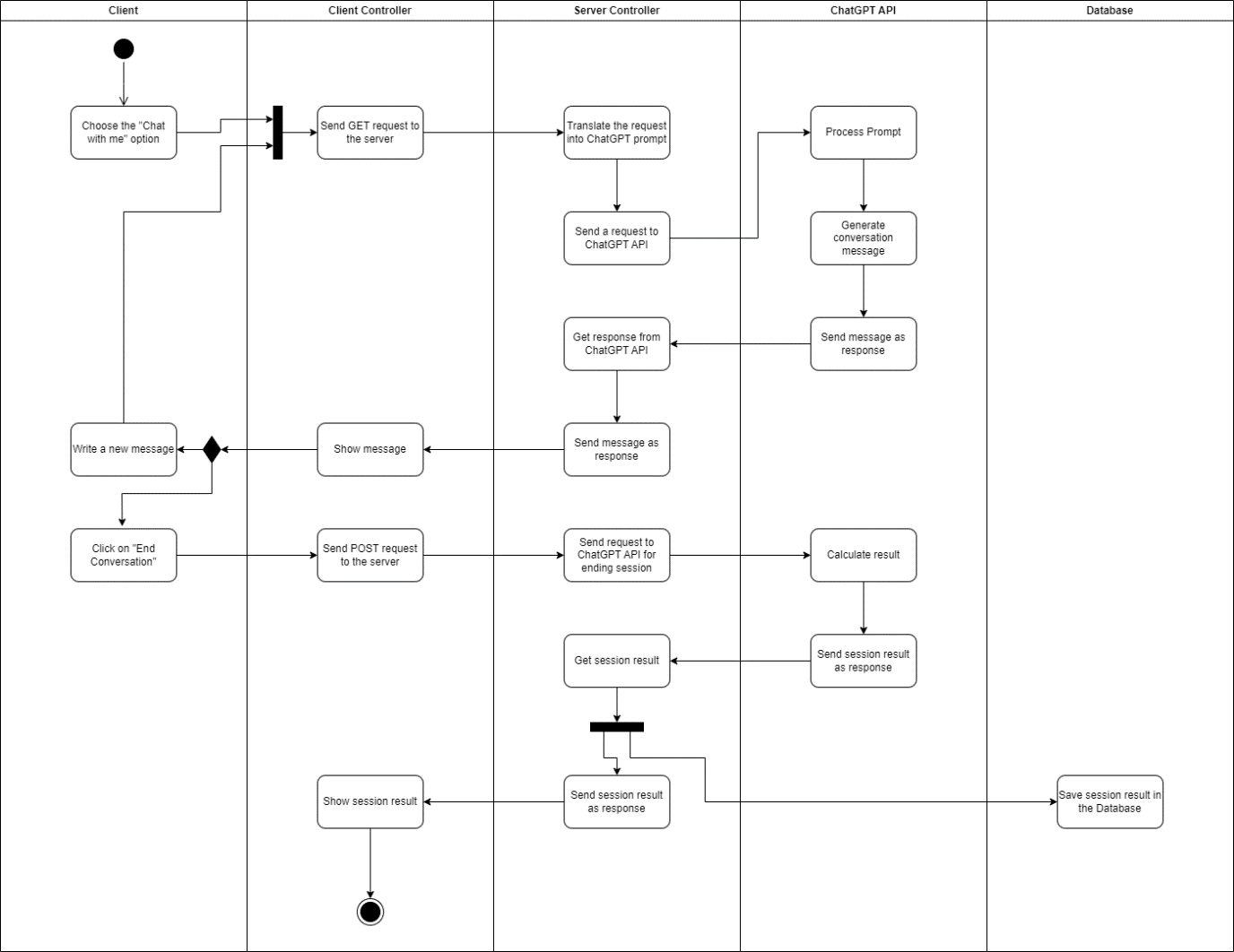


Figure 13 Activity Diagram - chat session

### Package Diagram

A screenshot of a computer screen

Description automatically generated

Figure 14 Package Diagram

### Deployment Diagram

A diagram of different types of software

Description automatically generated

Figure 15 Deployment Diagram

# Verification Plan

To verify the functionality of our application, we’ve devised a list of basic tests that will evaluate the implemented features of our program, which are based on our list of functional and non-functional requirements. We will periodically rerun these tests as we progress in the development process.

|  |  |  |
| --- | --- | --- |
| **Test Name** | **Description** | **Expected Result** |
| Successful Sign Up | The user clicks the sign up button and enters the required accout info correctly | The system validates the information and inserts the new user's data into the database |
| Unsuccessful Sign Up | The user enters incorrect account details | The system displays a message indicating that certain account details are incorrect |
| Successful Login | The user clicks the login button and enters the details of an account that exists in the database | The system validates the user’s credentials and redirects him to the home page |
| Unsuccessful Login | The user enteres the wrong credentials | The system displays a message indicating that the login data is incorrect |
| Logout functionality | The user clicks the logout button | The system successfully logs the user out |
| Sidebar navigation | The user clicks a nav item on the sidebar | The user is redirected to the relevant page |
| Chat mode functionality | The user enters the “chat with me” page and starts a conversation | The user successfully completes a conversation with the ChatGPT chatbot |
| Learn mode functionality | The users enters the “learn” page and starts a lesson | The user completes a full lesson |
| Leaderboard functionality | The user enters the “leaderboard” page | A list of high scores by fellow users is displayed on the page |
| My Profile functionality | The user enters the "My Profile" page | Displays the details of the user currently logged in |
| About page functionality | The user enters the “About” page | Displays technical information about the application |

# References

Achiam, J. A. (2023). *Gpt-4 technical report, page 6.* arXiv preprint.

Achiam, J. A. (2023). *Gpt-4 technical report, page 8.* arXiv preprint.

Angeles Sánchez-Elvira Paniagua, O. S. (2018). Developing Student Support for Open and Distance Learning: The EMPOWER Project. *Journal Of Interactive Media Education*.

Angular.io. (2023, September 25). *Introduction to Angular concepts*. Retrieved from https://angular.io/guide/architecture

Awati, R. (2023). *Cross-platform mobile development*. (Techtarget) Retrieved from https://www.techtarget.com/searchmobilecomputing/definition/cross-platform-mobile-development

Bastian, M. (2023). *GPT-4 has more than a trillion parameters - Report*. (The Decoder) Retrieved from https://the-decoder.com/gpt-4-has-a-trillion-parameters/

BrainStation. (2024). *What Is Web development?* (BrainStation) Retrieved from https://brainstation.io/career-guides/what-is-web-development

DB-Engines. (2024). *DB-Engines Ranking*. Retrieved from https://db-engines.com/en/ranking

Duolingo. (2024). *Duolingo*. Retrieved from https://www.duolingo.com/approach

Foster, L. (2023). *Learning Express.js - When You Don't Know Sh#t.* Career Kick Start Books, LLC.

G., M. R. (2022). *LANGUAGE AND ECONOMIC DEVELOPMENT.* Tashkent State Transport University.

Grosjean, F. (2019). A journey in languages and cultures: The life of a bicultural bilingual. Oxford University Press, USA.

Herron, D. (2020). *Node.js Web Development Fifth Edition.* Packt Publishing.

Marques-Schafer, G. &. (2018). *Languages learning conceptions and Duolingo: A critical analysis on its proposals and learners experiences.* Linguagem e Tecnologia.

Neil Cowie, K. S. (2019). Enhancing student retention rates on open non-formal online language learning courses. *Pacific Journal of Technology Enhanced Learning*.

OpenAI. (2022). *Introducing ChatGPT*. (OpenAI) Retrieved from https://openai.com/blog/chatgpt

OpenAI. (2023). *ChatGPT Plugins*. (OpenAI) Retrieved from https://openai.com/blog/chatgpt-plugins

Overflow, S. (2023, May). *2023 Developers Survey*. (Stack Overflow) Retrieved from https://survey.stackoverflow.co/2023/

Panchal, B. (2024). *Node.js Statistics: What They Tell Us About the Future of Web Development*. (Radix) Retrieved from https://radixweb.com/blog/nodejs-usage-statistics

Patricio Cerda Mardini, M. S. (2023). *a Comparative Analysis of Leading Large Language Models*. (mindsDB) Retrieved from https://mindsdb.com/blog/navigating-the-llm-landscape-a-comparative-analysis-of-leading-large-language-models

Sharma, A. (2023). *Express JS Tutorial*. (Simplilearn) Retrieved from https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js

Shawn Loewen, D. C. (2019). *Mobile-assisted language learning: A Duolingo case study.* Cambridge University Press.

Shortt, M. e. (2023). Gamification in mobile-assisted language learning: A systematic review of Duolingo literature from public release of 2012 to early 2020. *Computer Assisted Language Learning 36.3*, 517-554.

Wu, T. e. (2023). A brief overview of ChatGPT: The history, status quo and potential future development. *Journal of Automatica Sinica 10.5*, 1122-1136.