Ekaterina Seikkinen

How to Use ChatGPT for Programming

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communications Technology

Bachelor's Thesis

26 May 2023

Abstract

Author:	Ekaterina Seikkinen
Title:	How to Use ChatGPT for Programming
Number of Pages:	35 pages
Date:	26 May 2023
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communications Technology
Professional Major:	Mobile Solutions
Supervisor:	Miikka Mäki-Uuro, Senior Lecturer

The main goal of this thesis was to learn how to use ChatGPT for programming and create a mobile application project using an AI language assistant as a help tool. Proper use of ChatGPT in the programmer's workflow can reduce the time required to solve certain tasks. The thesis was aimed not only at developing application but at exploring the potential benefits and limitations of using intelligent chatbot as a development tool and figuring out the best ways to use it effectively.

Also, this thesis goes through the basic concepts of ChatGPT and explores different language models, which are used to build neural conversational models and their applications.

As a result of this thesis, a basic version of the Task Management application was developed, which included all the functionalities that were desired. The idea of the application is to help people to manage daily tasks to save time. Frontend was implemented with the help of React Native library and Authentication was implemented with the help of Firebase.

The process of creating the prototype went smoothly without any big issues. ChatGPT was especially helpful for generating code snippets, although the intelligent chatbot could not always solve problems or errors that came up. But overall, ChatGPT helped to build mobile application with basic functionality in few days. With experience in React programming and Firebase Authentication, understanding errors and challenges became easier. To take full advantage from AI language assistant some knowledge around programming is required.

Keywords: React Native, mobile development, ChatGPT, AI

Tiivistelmä

Tekijä:	Ekaterina Seikkinen
Otsikko:	ChatGPT:n käyttö ohjelmoinnissa
Sivumäärä:	35 sivua
Aika:	26.5.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Mobile Solutions
Ohjaaja:	Lehtori Miikka Mäki-Uuro

Opinnäytetyön tarkoituksena oli luoda mobiilisovellus käyttämällä tekoälyn kieliassistenttia apuvälineenä. ChatGPT:n oikea käyttö ohjelmoijan työskentelyssä voi lyhentää tiettyjen tehtävien ratkaisemiseen kuluvaa aikaa. Opinnäytetyön tarkoituksena oli sovelluksen kehittämisen lisäksi selvittää älykkään chatbotin käytön mahdollisia etuja ja rajoituksia kehitystyökaluna ja selvittää parhaat tavat käyttää sitä tehokkaasti.

Yksi tekoälyn alueista ovat kielimallit. Kielimalli on todennäköisyysjakauma sanoihin tai sanasarjoihin. Kielimalleja käytetään monissa palveluissa, esimerkiksi ne auttavat puheassistentteja yläpitämään dialogeja, antavat nopeita vastauksia haussa, parantavat käännösten laatua ja paljon muuta. Yksi edistyneimmistä kielimalleista on OpenAI:n kehittämä GPT-3. GPT-3:ssa on 175 miljardia parametria, ja sen harjoittamiseen käytettiin valtava määrä tekstiä. ChatGPT on erittäin suosittu tekoälypohjainen ohjelma, jota käytetään dialogien luomiseen. Älykkäässä chatbotissa on kielipohjainen malli, jota kehittäjä hienosäätää ihmisten välistä vuorovaikutusta varten.

Opinnäytetyön tuloksena kehitettiin Task Management-sovelluksen perusversio, joka sisälsi kaikki halutut toiminnot. Sovelluksen idea on auttaa ihmisiä hallitsemaan päivittäisiä tehtäviä ajan säästämiseksi. Käyttöliittymä toteutettiin React Native -kirjaston avulla, ja todentaminen toteutettiin Firebase-kehitysalustan avulla.

Prototyypin kehityksessä ei ilmennyt suurempia ongelmia. ChatGPT oli erityisen hyödyllinen koodinpätkien luomisessa, vaikka älykäs chatbot ei aina pystynyt ratkaisemaan esiin tulevia ongelmia tai virheitä. React-ohjelmointiin ja Firebase-todennukseen perehtyminen auttoi virheiden ja haasteiden ymmärtämisessä. Tekoälyn kieliassistentin täyden tehon hyödyntäminen edellyttää ohjelmoinnin tuntemusta.

Avainsanat: React Native, mobiilikehittäminen, ChatGPT, tekoäly

Contents

List of Abbrevations

AI:	Artificial Intelligence
GPT:	Generative Pre-Trained Transformer
IDE:	Integrated Development Environment
NLP:	Natural Language Processing
RLHF:	Reinforcement Learning from Human Feedback
RM:	Reward Models
SFT:	Supervised Fine-Tuning

1 Introduction

Everything changes and modern technology is in demand in all areas of life. Neural networks and artificial intelligence make human life easier by solving many problems several times faster than humans. ChatGPT launched by the company OpenAI on November 30, 2022, impressed the IT community with the accuracy of its answers to certain questions, becoming the most discussed project in the field of artificial intelligence. ChatGPT is a chatbot that uses artificial intelligence, and its main features include answering any question, inventing stories, asking additional questions, translating texts, giving advice, and helping with programming.

The advent of ChatGPT raises important questions about how many people's lives will change and whether professionals will lose their jobs. The main purpose of the thesis project is to study the use of ChatGPT in programming and to check the accuracy and correctness of its answers, as well as to understand how it can simplify the programmer's workflow. The engineering work also examines the language models used to create neural conversational models and explores the basic concepts of ChatGPT.

The work first introduces the technologies related to ChatGPT, including its history of creation and prototypes, and then demonstrates its use in programming by providing an example of creating a mobile application.

The idea for the engineering job came from Suomen Sijoitutkimus Oy, the company where I work. The correct use of ChatGPT in a programmer's work can shorten the time required to solve certain problems and become an indispensable assistant at work.

Chapter 2 provides an overview of artificial intelligence and its basic concepts, including the history of the creation of ChatGPT and key elements of intelligent

chatbot. In chapter 3, the thesis goes through the details of case project, exploring the development process of a mobile application using a conversational AI. Chapter 4 aims to provide practical insights and best practices for utilizing Chat GPT for application development.

2 Introduction to ChatGPT

The chapter begins with an overview of artificial intelligence, its definition, and key technique such as natural language processing. This is followed by an explanation of language models, including their purpose, structure, and types. After that, chapter introduces the history of ChatGPT, its creators, the process of development and fundamental principles of artificial language model.

2.1 Artificial Intelligence

Artificial intelligence has permeated every aspect of modern life and continues to gain interest. AI refers to the simulation of human intelligence in machines and is based on the principle that human intelligence can be defined in a way that allows machines to imitate it and perform tasks. (Frankenfield 2022). It can be a manifestation of some kind of creative ability, a tendency to reason, to generalize, to learn from early experience.

John McCarthy (the founder of functional programming and the inventor of the Lisp programming language) first mentioned the term "artificial intelligence" in 1956. However, Alan Turing formed the idea of such a system in 1935. The scientist created a description of an abstract computer consisting of unlimited memory and a scanner that moves back and forth through the memory. The earliest successful artificial intelligence program was created by Christopher Strachey in 1951, and already in 1952 it played checkers with a human and surprised the public with its ability to predict movements (History of Artificial Intelligence 2020). Nowadays, such programs are developing very quickly due to the digitization of information and the growth of its turnover and volume.

AI algorithms are at the core of artificial intelligence, which exhibit some of the behaviours associated with human intelligence. These include computer-enhanced learning, reasoning, perception, and imitating human cognitive activity (History of Artificial Intelligence 2020). AI systems work by analysing large amounts of labeled training data for correlations and patterns, which are used to predict future states. For example, an AI language assistant that is fed text chat examples can learn to produce a realistic conversation with people through Natural Language Processing (NLP) (Roldós 2020).

There are two types of artificial intelligence: strong AI and narrow AI. Strong AI refers to programming that can replicate the cognitive abilities of the human brain, such as cyber security. Narrow AI, on the other hand, is an AI system designed and trained to perform a specific task, such as Apple's Siri.

AI is widely used in various fields from finance to healthcare, and it has become integrated into everyday devices, such as vacuum cleaners. Examples of AI include chatbots, intelligent assistants, robo-advisors, email spam filters, and Netflix recommendations.

Nowadays artificial intelligence is used in various fields from finance to healthcare, and it is also integrated into the most common facilities, such as the vacuum cleaner. The inner workers of artificial intelligence are not fully understood, but it is predicted by experts that the development of artificial intelligence will come even closer to the development of the human brain within the upcoming years. (Frankenfield 2022).

2.2 Language Models

A language model is a core component of modern Natural Language Processing (NLP). It is a statistical tool that analyses the pattern of human language to predict words. NLP is a branch of artificial intelligence that allows machines to understand human language. Its goal is to build systems that

understand text and automatically perform tasks such as translation, spelling, or topic classification. (Roldós 2020).

The language model uses machine learning to perform a probability distribution on the words used to predict the most likely next word in the sentence based on the previous entry. Simply put, a language model is an algorithm that, based on n words of the proposed text, can guess what the next word is and what the probability of this word's occurrence is. The language model always receives text as input and produces text data as output. Since many tasks can be formulated in the text, language models are used extensively. (Kapronczay 2022).

Working with language models is based on probability theory: algorithms calculate the probability that a certain word appears in the text. For this, context, style of speech and meaning of words must be considered, but the machine can only work with numbers. Therefore, the text is converted into its numerical representation - this process is called encoding, and the result is a numerical sector. Each word has a numerical representation, and the neural network looks at which word combinations and in which order are most often found together in the language. The larger and more diverse text set is taught to it, the more complex dependencies the model captures and repeats them with new information. (Kapronczay 2022).

There are two types of language models:

1. Probability-based methods

The language model examines the word's probability of occurrence based on text examples. Simpler models can consider the context of a short string of words, while larger models can work at the sentence or paragraph level.

2. Neural network-based modern language models

Neural network-based language models ease the sparsity problem by encoding inputs. Word embedding layers create a vector of arbitrary size from each word,

which also contains semantic relations. These continuous vectors create the desired precision in the probability distribution of the next word. (Kapronczay 2022).

The most famous language models are GPT-3 (developers: OpenAI, parameters: 175 billion), Bloom (developers: Hugging Face, BigScience; parameters: 176 billion), ESMFold (developers: Meta AI, parameters: 15 billion), Gato (developers: DeepMind, parameters: 79 million, 364 million and 1.18 billion), LamDa (developers: Google, parameters: 137 billion). The number of parameters in a language model is a measure of its ability to recognize complex patterns and relationships in data and develop new emergent capabilities. (Wodeski 2022).

2.3 OpenAI

OpenAI is an artificial intelligence research laboratory and development company. It has created a variety of AI-powered programs and machine learning algorithms that allow computers to do all kinds of things, such as create images from text or make a robotic hand that solves Rubik's cubes (Ot 2022). This altruistic AI company was founded by Elon Musk and Sam Altman in 2015. It is officially a non-profit organization and works with funds invested by investors. The company is directly trying to build safe and useful artificial intelligence (AGI), but also considers its mission accomplished if the work helps others achieve that result. (A Guide to OpenAI and its Products and Services 2023).

OpenAI conducts research in many fields related to artificial intelligence, including machine learning, robotics, economics, and computing. In addition to research work, OpenAI also aims to educate the public about artificial intelligence and its possible effects and to promote the responsible development and use of artificial intelligence.

During its seven years of existence, OpenAI has developed many artificial intelligence tools that perform specific tasks. They are as follows:

1. OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms.

2. OpenAI RoboSumo is a simulated environment for developing and testing robot control algorithms.

3. The OpenAI Debate Game simulates a debate between two or more participants using artificial intelligence.

4. OpenAI Dactyl is a robotic hand designed to learn and adapt to new tasks using machine learning algorithms.

5. OpenAI Generative models (including DALL-E and ChatGPT) are artificial intelligence systems trained to generate new, original content based on a set of examples or a specific model. (A Guide to OpenAI and its Products and Services 2023).

2.4 GPT-3 Language Model

GPT-3 is Generative Pre-trained Transformer 3, and it is the third version of the language model released by OpenAI in May 2020. It is generative because GPT-3 can produce long sentences from unique text as output. Most neural networks can only produce yes or no answers or simple sentences. Pretrained means that the language model is not built with domain-specific knowledge but is able to perform domain-specific tasks such as translation. Transformer is a neural network architecture developed by Google researchers in 2017. (What is GPT-3, How Does It Work, and What Does It Actually Do? 2021).

Deep learning can be considered as a specialized branch of machine learning. It is a field that relies on learning and improving on its own by examining computer algorithms. While machine learning uses simpler concepts, deep learning works with artificial neural networks, which are designed to imitate how humans think and learn. (Reyes 2023).

GPT-3 is a deep learning-based language model that can generate text that closely resembles human-written content. In addition to text, it can also produce code, stories, poems, and other forms of creative content. As a result of these capabilities, it has gained significant attention and popularity in the field of natural language processing. The full version of OpenAI GPT-3 is, with approximately 175 billion parameters to be trained, the largest model trained so far compared to other language models. (Cooper 2021).

The GPT-3 model differs from other language models in that it has a huge number of parameters to train, 10 times more than any previous model. In general, the more parameters a model has, the more data is needed to train the model. According to the authors, the OpenAI GPT-3 model has been trained with around 45 terabytes of text data from various sources such as Wikipedia and books. (Cooper 2021).

GPT-3 is not a single model but a family of models. Each model in the family has a different number of parameters to be trained. The table in Figure 1 shows each model, architecture, and corresponding parameters.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Sizes, architectures, and learning hyper-parameters of the GPT-3 models

Figure 1. An example of the GPT-3 models family (Cooper 2021).

The OpenAI GPT-3 model family is based on a transformer-based architecture (Figure 2).

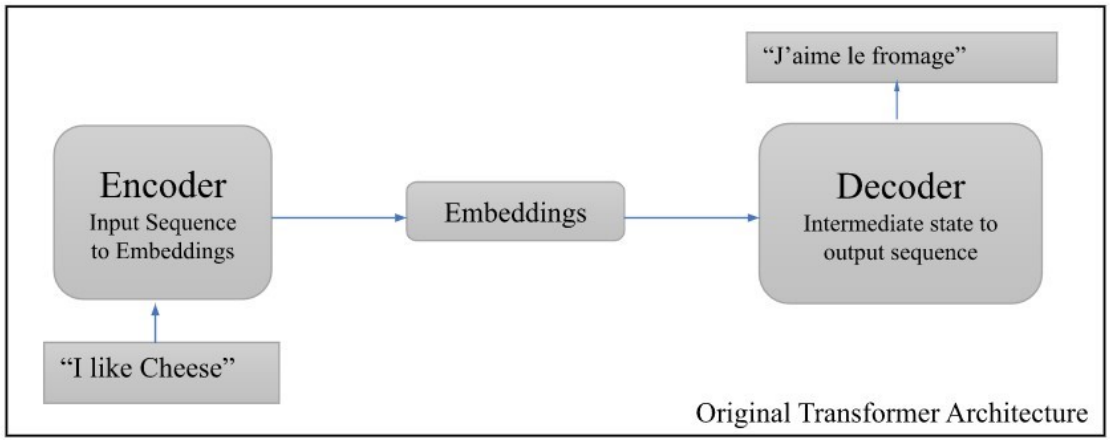


Figure 2. Transformer-based architecture (Cooper 2021).

Encoder: The encoder receives the input and constructs a representation (its features) from it. This means that the model is optimized to understand the input.

Decoder: The decoder uses the encoder representation (features) along with other inputs to generate the target sequence. This means that the model is optimized for output.

A key part of transformers is the attention process, which allows the model to understand which words are most important to consider when judging what the sequence produces. In each iteration of the encoder, each word is given numerical weights, which are then analysed by the decoder to determine key aspects of the input text on which the output should be based. (Bard 2021).

2.5 Instruct GPT

Instruct GPT, or simply Instruct, is a powerful tool that allows users to fine-tune the generation features of the GPT model language. Developed by OpenAI, Instruct GPT allows users to train a model for specific tasks and generate text personalized to their specific needs. This tool is designed to facilitate developers and researchers fine-tuning GPT for a wide range of natural language processing (NLP) tasks such as language translation, text summarization, and

question answering. Instruct GPT allows users to train the model with their own data, fine-tune the model for specific use cases, and create text that is relevant and accurate. The goal of Instruct GPT is to facilitate and improve the automation of repetitive and time-consuming tasks of companies and organizations. (Dongre 2023).

The key difference between GPT and Instruct GPT is that GPT can be fine-tuned for any task, while Instruct GPT needs to be specifically fine-tuned for a specific set of tasks. While Instruct GPT is highly proficient at certain tasks, its narrower focus makes it less adaptable than the original GPT.

Compared to GPT-3, the new Instruct GPT models adhere better to English instructions, are less likely to produce false information, and at least slightly less likely to produce harmful results.

One issue with GPT-3 is that it was trained to predict the next word from a large data set and not safely perform the task the user wanted. To solve the problem, OpenAI used a method called reinforcement learning with human feedback (RLHF). Reinforcement learning is a technique that involves an AI agent making decisions based on the actions it performs in its environment and the feedback it receives in the form of rewards or punishments. (Cuofano 2023).

To convert GPT-3 models to Instruct GPT models, OpenAI designed a three-step procedure. The first is model fine-tuning. The second is building a reward model (RM). Third, the SFT (Supervised Fine Tuning) model must be taken and further fine-tuned using reinforcement learning. Instruct GPT is more attuned to human preferences, which ultimately better predicts actual performance. The reason is that Instruct GPT is better aligned with human intentions through a reinforcement learning paradigm that makes it learn from human feedback. (Lowe & Wainwright 2022).

2.6 ChatGPT

OpenAI's ChatGPT is a very popular AI-based program used to create dialogs. An intelligent chatbot has a language-based model that the developer fine-tunes for human interaction (Pocock 2023).

Instruct GPT and ChatGPT are both models developed by OpenAI. However, the biggest difference between them is in their intended use. ChatGPT is a conversational AI model trained to generate human-like responses in dialogue. On the other hand, Instruct GPT is a task-oriented AI model that is trained to perform specific tasks such as code generation, text SQL queries, and summarization by following instructions given to it.

Both models are built on top of transformer architecture but Instruct GPT is trained on more versatile tasks and is fine-tuned to perform specific tasks, while ChatGPT is trained on more general conversational data. Instruct GPT is a specialized version of ChatGPT designed for a task-oriented context. (Dongre 2023).

ChatGPT works by collecting human-written data from the Internet and using calculated predictions to answer questions and queries entered by the user. The responses it generates are based on text queries and information from which the AI language assistant "learns" more about various topics and how to discuss them. (Pocock 2023).

It is well trained on biased and unbiased data in the form of texts from books, articles, and websites. ChatGPT can reproduce output and reliability, which is critical for many sensitive applications and other valuable AI systems. However, it is still prone to error and relies on training data provided in 2021. (Pocock 2023).

3 Task Management Application

Chapter introduces the idea and goals of the case project, technologies, and tools that are used to create the application. After that, details of development process are provided through this chapter such as key components of the application, which are developed with React Native (Task List and Task Details), implementation of user authentication and authorization through Firebase as well as storing task details in Firestore database.

3.1 Purpose and Goals of the Application

During the search for the best application idea to implement for an IT project, ChatGPT provided 10 different options. The idea of creating a Task Management application was found to be the most appealing, as it has the potential to help individuals who often struggles with managing multiple tasks.

The purpose of the task management application is to provide platform to simplify the task management process and help users achieve their goals with greater ease and efficiency. With an increasing number of tasks to complete in our daily lives, it is becoming more challenging to stay organized and prioritize our tasks. This application solves this problem by providing users with an intuitive and user-friendly interface to manage their tasks.

The application will allow users to create new tasks and filter them based on their completion status. This will help users understand which tasks have been completed and which ones still need to be done.

The basic functionality of the Task Management Application has been successfully implemented with the help of ChatGPT. The application consists of five screens, including Authentication Screens for user login and registration, a Task List Screen that displays all the tasks in a list, a Task Form Screen for creating new tasks, and a Task Details Screen that allows users to view additional

details about a task, such as its description. The Task Details Screen also features a Delete button, which enables users to remove tasks from their task list.

3.2 Development Process: Technologies and Tools

The application was developed using React Native, TypeScript, Expo, Firebase Authentication, and Firebase Firestore. Both web and mobile applications are divided into frontend and backend. Frontend refers to the part visible to the user and its functions, while backend refers to the basis of the application, which the frontend is designed to utilize.

In a React Native application, the line between frontend and backend can be a bit blurry since much of the application logic and state management is handled on the client-side. However, Firebase Authentication and Firestore can be considered part of the backend since they are cloud-based services that provide authentication and data storage functionality.

Frontend:

- React Native: a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It is based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting to the browser, it targets at mobile platforms. In other words: web developers can now write mobile apps that look and feel truly native, all with the JavaScript library we already know and love. (Eisenman, 2015).

- TypeScript: a superset of features that enable us to use types in JavaScript. It gives us a module system, classes, interfaces, and a rich type of system. By using TypeScript write better code, with fewer errors, it helps to catch mistakes statically that can be committed along the way while applications are build. The code quality and understandability of our application can be enhanced just by starting to use types in our code. (Maldonado 2019).

- Expo: an open-source platform to build React Native apps. It is a set with tools and services built for React Native. Expo can help developers get started with building React Native apps easily. It offers a range of tools that simplify the creation and testing of React Native apps. Additionally, Expo provides a more robust and convenient development workflow with the flexibility to customize and configure project settings to suit individual needs. (Borozenets 2022).

Backend:

- Firebase Authentication: a service provided by Firebase that enables developers to authenticate users using email and password, phone number, or social media accounts such as Google, Facebook, and Twitter. A secure and easy way to authenticate users in the application is provided by it. (Firebase Authentication 2023).

- Firebase Firestore: a NoSQL cloud database provided by Google that allows developers to store and sync data between client-side and server-side applications in real-time. Store, synchronization, and query application data at global scale can be enabled by Cloud Firestore. (Cloud Firestore 2023).

3.3 Utilizing ChatGPT for Application Development

ChatGPT was used as a helpful tool throughout the mobile application development process. Various application ideas were received from ChatGPT initially, and then the AI language assistant was asked to determine the most suitable elements to include, taking into consideration that it was a study project (Figure 3). Valuable advice was provided by chat bot throughout the entire process.

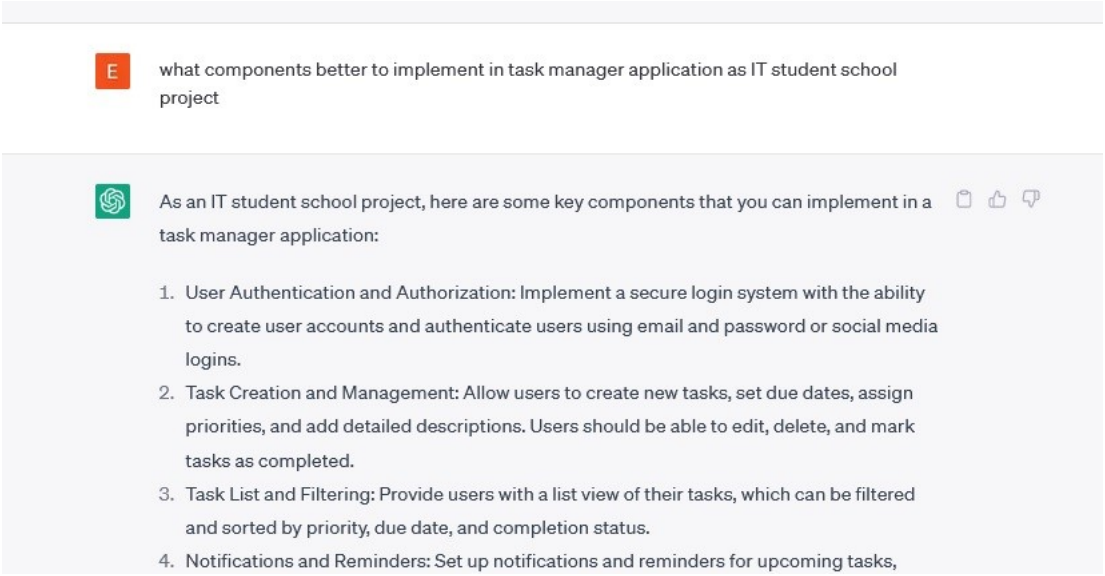


Figure 3. Appropriate components to include in study case.

This list can be utilized to create a product design map. It has been decided that first three components would be implemented in study case, only limited number of features in the first iteration of the product.

Chat bot also helped with the application folder structure (Figure 4) and with a choice of colour scheme (Figure 5).

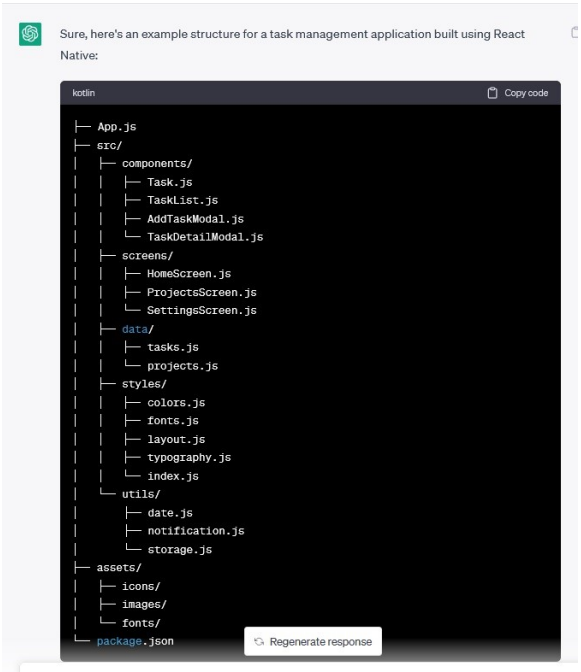


Figure 4. Possible Task Management application folder structure, suggested by ChatGPT.

The folder structure seems reasonable and logically organized.

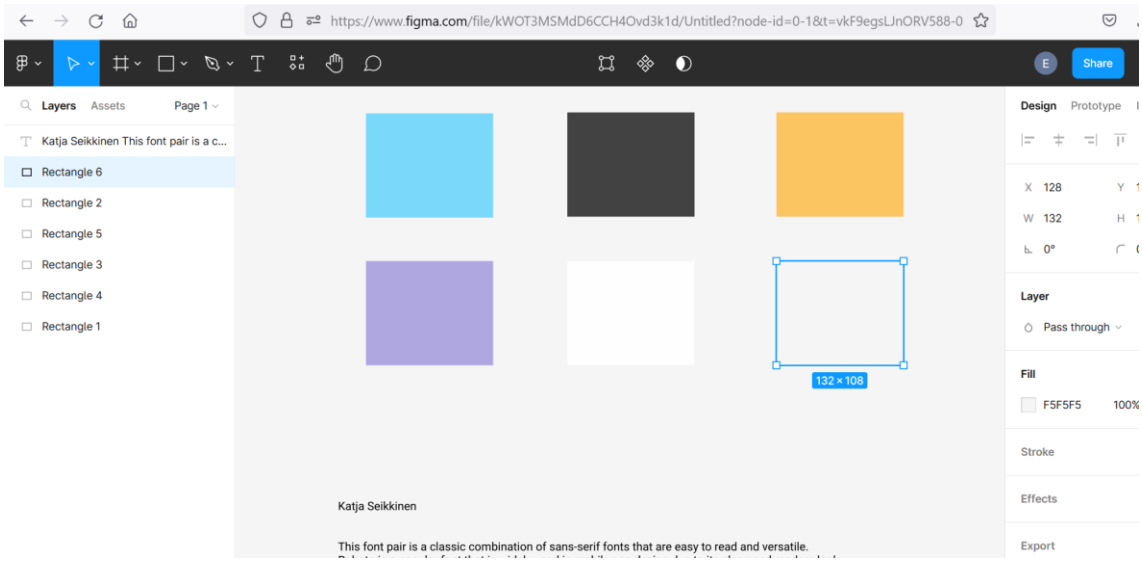


Figure 5. Possible colour schemes for Task Management application, suggested by ChatGPT (represented by Figma).

The colour schemes are created primarily from neutral colours, these schemes are helpful and seems suitable for the Task Management application.

In addition to that, the AI language assistant provided helpful guidance on how to initiate a new React Native project with TypeScript and Expo, including step-by-step instructions for setting up the development environment, configuring the project, and creating the necessary files and directories.

When the decision about the components to be used in the application was made, the ChatGPT was asked for assistance in creating and styling individual components (Figure 6).

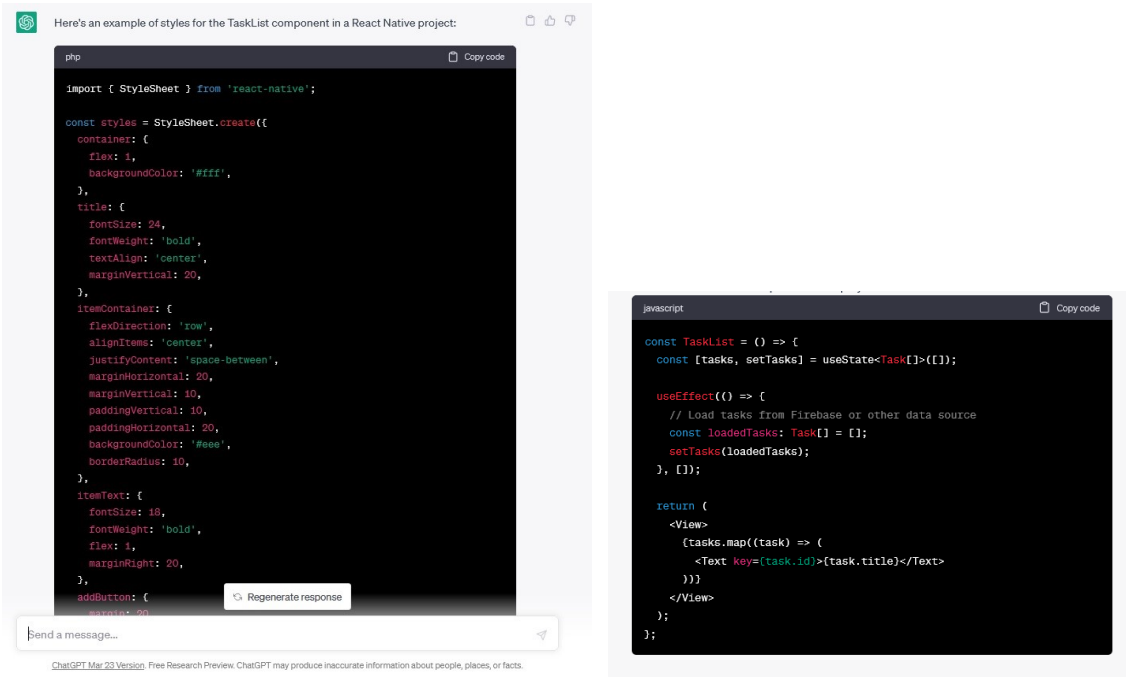


Figure 6. Example of creating individual component with ChatGPT on the example of Task List.

3.4 Components to be Implemented in the Application

The Task Management application was developed by creating a simple version with a few screens and basic functionality, and all screens, elements, styles and functions were created with the help of ChatGPT. In general, there are five screens in the study project: Sign In, Sign Up, Task List, Task Form, Task Details.

3.4.1 User Authentication and Authorization

The User Authentication and Authentication components are implemented through two screens, the Sign In and Sign Up screens (Figure 7). These screens are integrated with Firebase Authentication to provide a secure and efficient user authentication process.

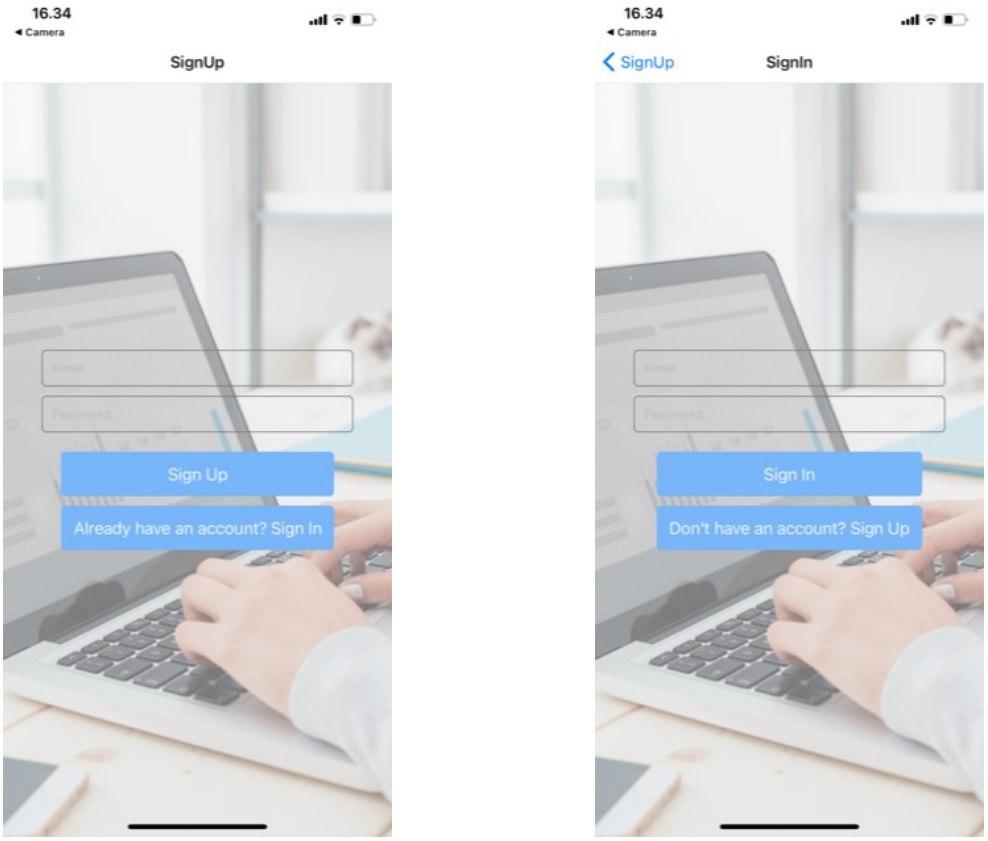


Figure 7: Sign In and Sign Up Screens of the Task Management application.

The implementation of the Authorization and Authentication of the application started with the choice of technologies, where it was decided to use third-party authentication service: Firebase Authentication. ChatGPT suggested a few more options such as: AuthO, AWS Cognito and Okta, but since I have an experience working with Firebase Authentication, the decision was clear.

ChatGPT helped to create new Firebase project for the React Native application with the comprehensive and detailed answer. Firebase Authentication offers a set of backend services, SDKs, and pre-built UI libraries to authenticate users to the application, and the email/password sign-in method was chosen for this project. Firebase Authentication has intuitive and user-friendly interface (Figure 8).

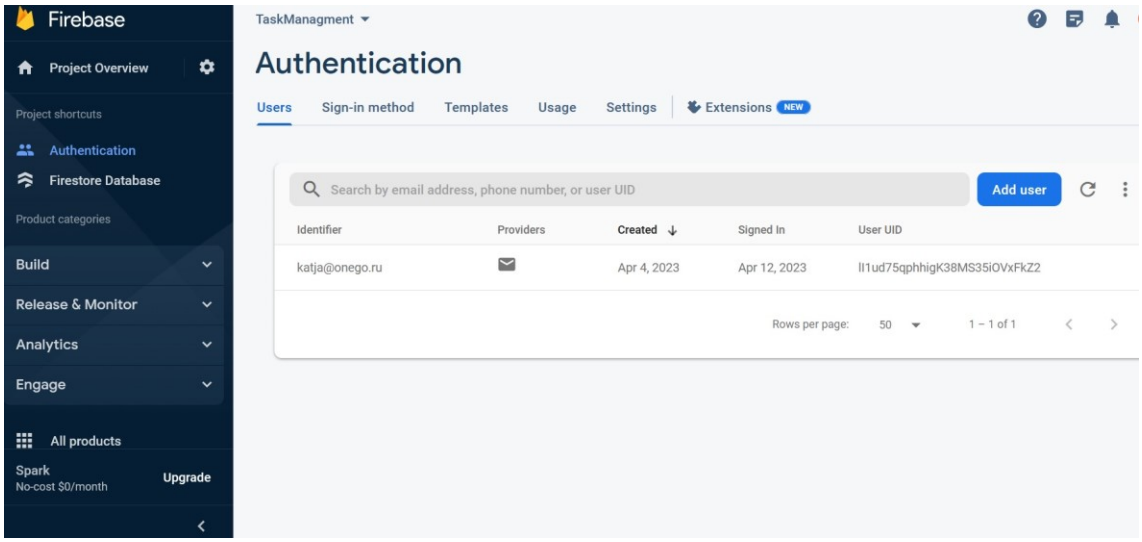


Figure 8: Firebase Authentication user interface

After Firebase project with the Authentication was created, ChatGPT was requested to provide guidance on implementing Sign In and Sign Up screens in React Native using Type Script with the Firebase Authentication. The main step to initialize a Firebase project in a React Native mobile application is to create a configuration object that contains the following fields: apiKey, authDomain, projectId, storageBucket, messagingSenderId, and appId. This information can be easily obtained from Firebase Console.

The chatbot generated code for Sign In and Sign Up screens can be easily integrated into the React Native project by copying and pasting the code (Figure 9).

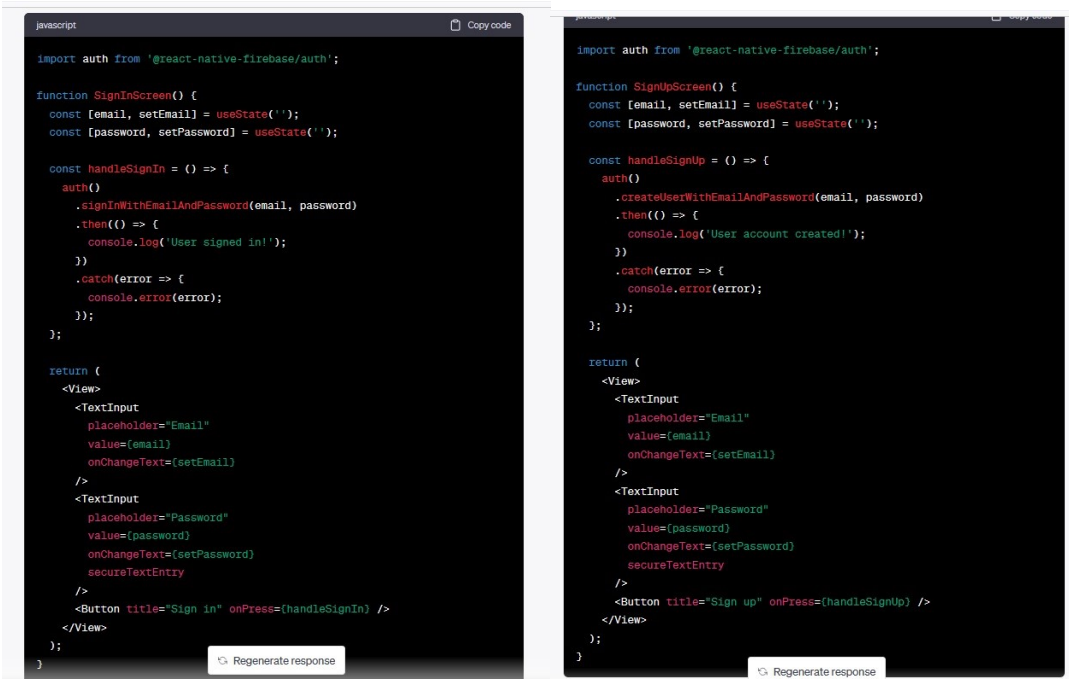


Figure 9. Sign In and Sign Up screen implementation in React Native generated by ChatGPT.

The solution generated by ChatGPT for Sign In and Sign Up screens implementation in React Native has some disadvantages. One of them is that it does not include import statements for the useState() React hook, which need to be added manually in order for the code to work properly. Second one is the auth() method, imported from react-native-firebase, is not compatible with Expo, which I used for a smoother and faster development process. As a result, I had to use the "firebase" library instead of "react-native-firebase". To avoid such issues in the future, it is important to provide ChatGPT with more detailed information about the project, such as how to implement Sign In and Sign Up screens in React Native with TypeScript and Expo.

I encountered another problem while trying to initialize the Firebase project in my React Native application. The solution provided by ChatGPT was to use the command "firebase.initializeApp(firebaseConfig)" to initialize the Firebase project in the React Native application. But there was an error: “TypeError: Cannot read property 'initializeApp' of undefined”. Unfortunately, the Chatbot was unable to provide a solution for this issue, so I had to search for answers online

through Google. The decision was to refactor the code (Figure 10), when use **Firebase Modular API (v9**), which is the newest version available.

```
// Initialize Firebase
const app: FirebaseApp = initializeApp(firebaseConfig);
const auth = getAuth(app)
const db = getFirestore(app)
```

Figure 10. Firebase initialization in React Native project (Firebase Modular API (v9)).

After requesting assistance from chatbot to implement Sign In and Sign Out screens, it provided recommendations for using Navigation in React Native to improve the user experience of the application. This suggestion was very useful.

The only issue with the advice provided by the chatbot on how to use Navigation in React Native was that it didn't include all the necessary code parts for the implementation. As a result, when the code was copied and pasted into the application project, an error occurred (“Cannot find name RootStackParamList”). It was necessary to seek further information from the AI language model on how to resolve this error (Figure 11).

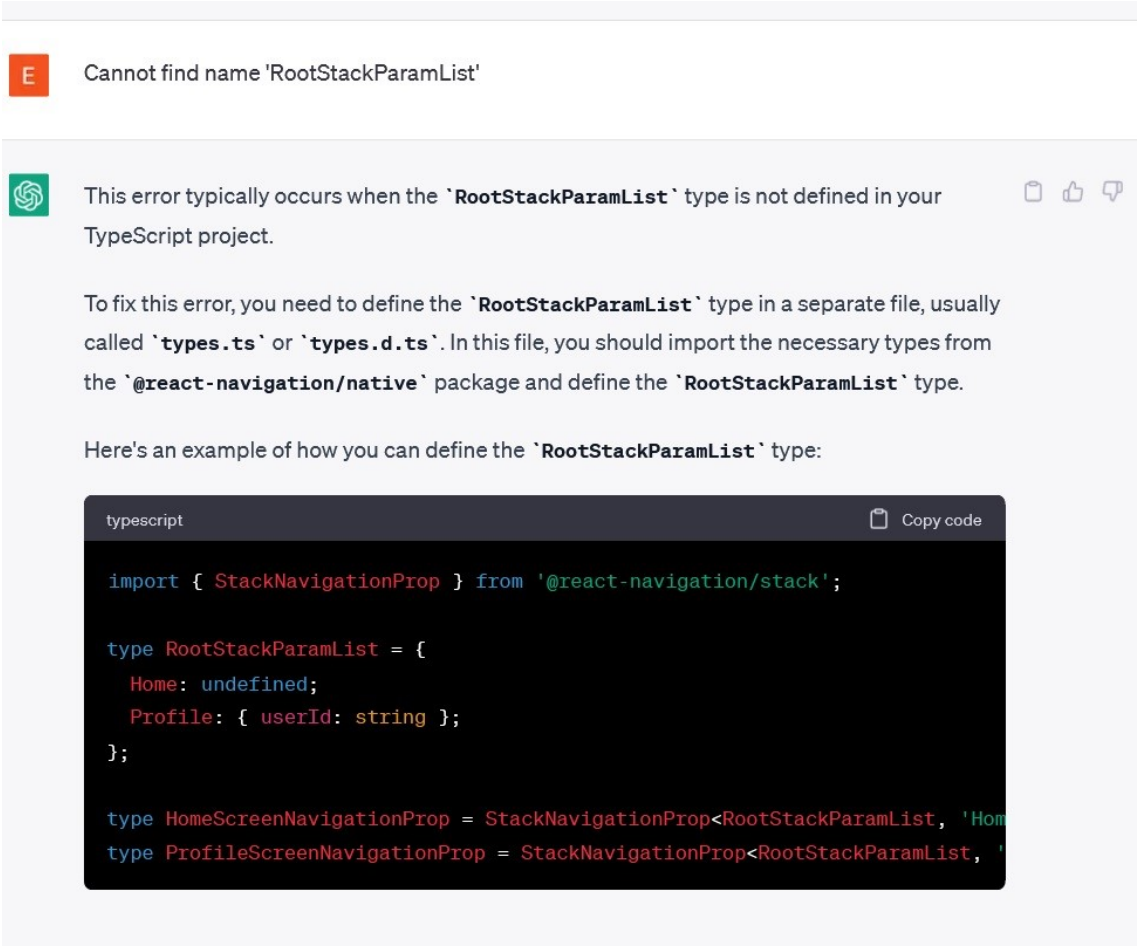


Figure 11. Prompt example.

The Chatbot also assisted with styling the Sign In and Sign Up screens. It offered useful advice, such as using a simple colour scheme, ensuring that form fields are easy to read, using contrasting colours for buttons, making sure buttons are large enough, and implementing responsive design.

3.4.2 Task Creation and Management

Task creation is implemented through the Task Form screen (Figure 12). The Task Form screen is a key component of the Task Management application that allows users to add new tasks to their task list. Task Form consists of two Text Input components for entering task item and task description, as well as an Add Task button for submitting the task. When the user submits a new task using the Add Task button, the task information is sent to Firestore and stored as a

new document in a 'tasks' collection. Each task document contains fields for the task item and description, as well as metadata such as the creation date. This data can then be retrieved and displayed in the Task List screen, allowing users to see all their tasks.

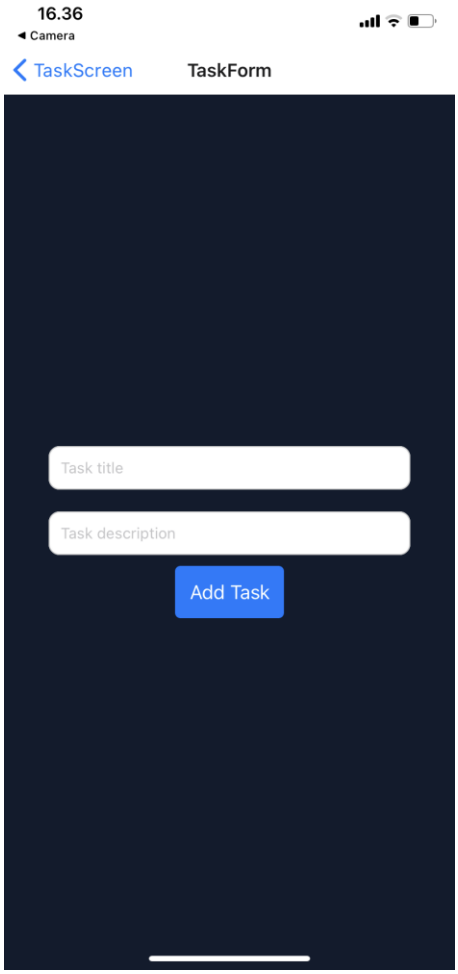


Figure 12. Create Task screen.

During the development of the Task Form screen, assistance was requested from ChatGPT to help with writing the code to add items to a Firestore collection.

Code example to add task to Firestore with React Native and TypeScript (Figure 13). In this example we are using db (const db = getFirestore(app)) with getFirestore() function from “firebase/firestore” library. Then, we're calling the addDoc() method on the “task”collection to add a new document with the task

name, task description and creation timestamp. If the operation is successful, we log a success message to the console. If there's an error, we log an error message instead.

```
const handleTaskSubmit = async (task: Task) => {
  try {
    await addDoc(collection(db, 'tasks'), { task: task });
    console.log('Task added successfully');
  } catch (error) {
    console.error('Error adding task: ', error);
  }
};
```

Figure 13. Code snippet to add task to Firestore.

Cloud-based NoSQL document database Firestore provides real-time updates and synchronization between different devices and platforms, allowing for seamless collaboration and data sharing. The chatbot provided clear instructions on how to create a Firestore collection within a Firebase application. The Firestore interface is designed to be intuitive and user-friendly (Figure 14).

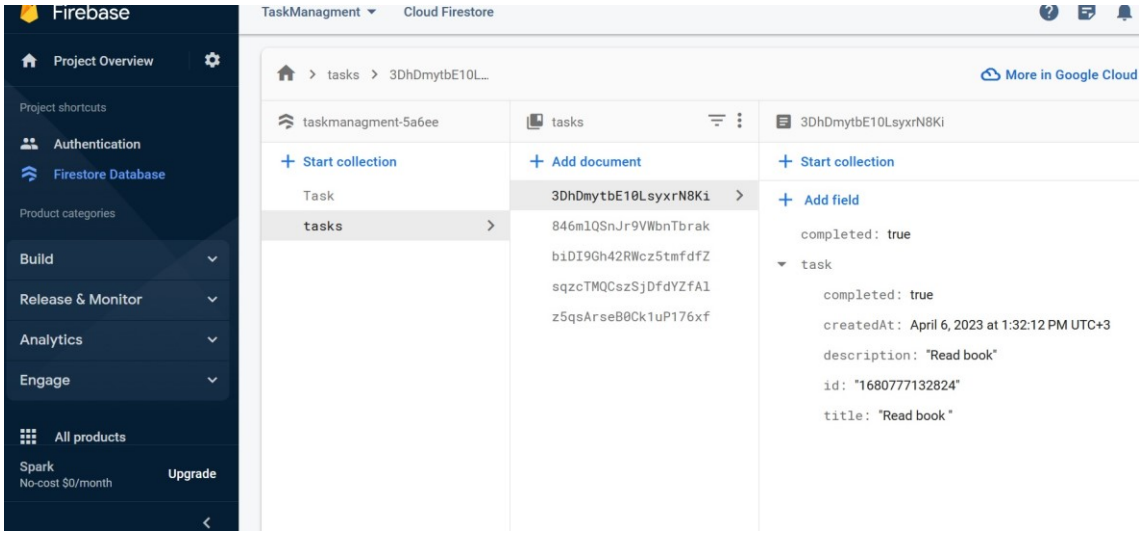


Figure 14. Firestore database user interface.

3.4.3 Task List and Filtering

The Task List (Figure 15) screen is a one of the most important components of the Task Management application that displays all the user's tasks in an organized and easily accessible manner. This screen includes a list of all the user's tasks and simple filter that allows the user to easily view their completed and uncompleted tasks. It also allows users to navigate to the Task Details screen by clicking on a specific task. Completed tasks are displayed with a blue background while uncompleted tasks are displayed with a white background on the screen. This helps users easily identify the status of their tasks. By clicking the plus button in the right bottom corner on the Task List screen, users can navigate to the Task Form screen to create a new task. Task List screen provides an efficient and user-friendly way for users to manage their tasks.

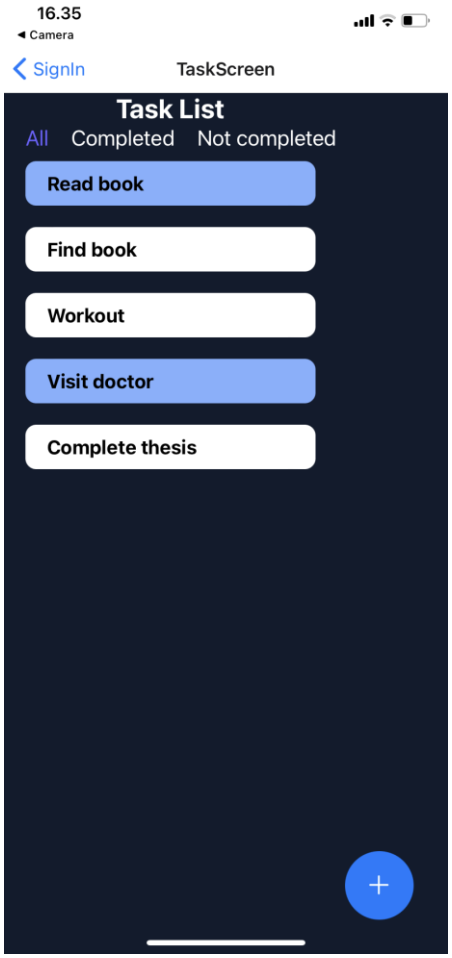


Figure 15. Task List screen.

The task items displayed in the Task List screen are fetched from Firestore, providing real-time updates to the user's task list.

There is a React Hook within filter function (Figure 16). This Hook is triggered whenever the status of the filter is changed and is responsible for returning a filtered list of tasks based on the selected filter status. The “useEffect” Hook is a built-in React Hook that enables developers to perform side effects after a component has been rendered. In this case, the Hook is used to apply the appropriate filter to the list of tasks based on the value of the “filterStatus” variable.

By utilizing the “useEffect” Hook, the filter function is automatically triggered whenever the value of the “filterStatus” variable is updated. This ensures that the filtered list of tasks is always up-to-date and reflects the current filter status.

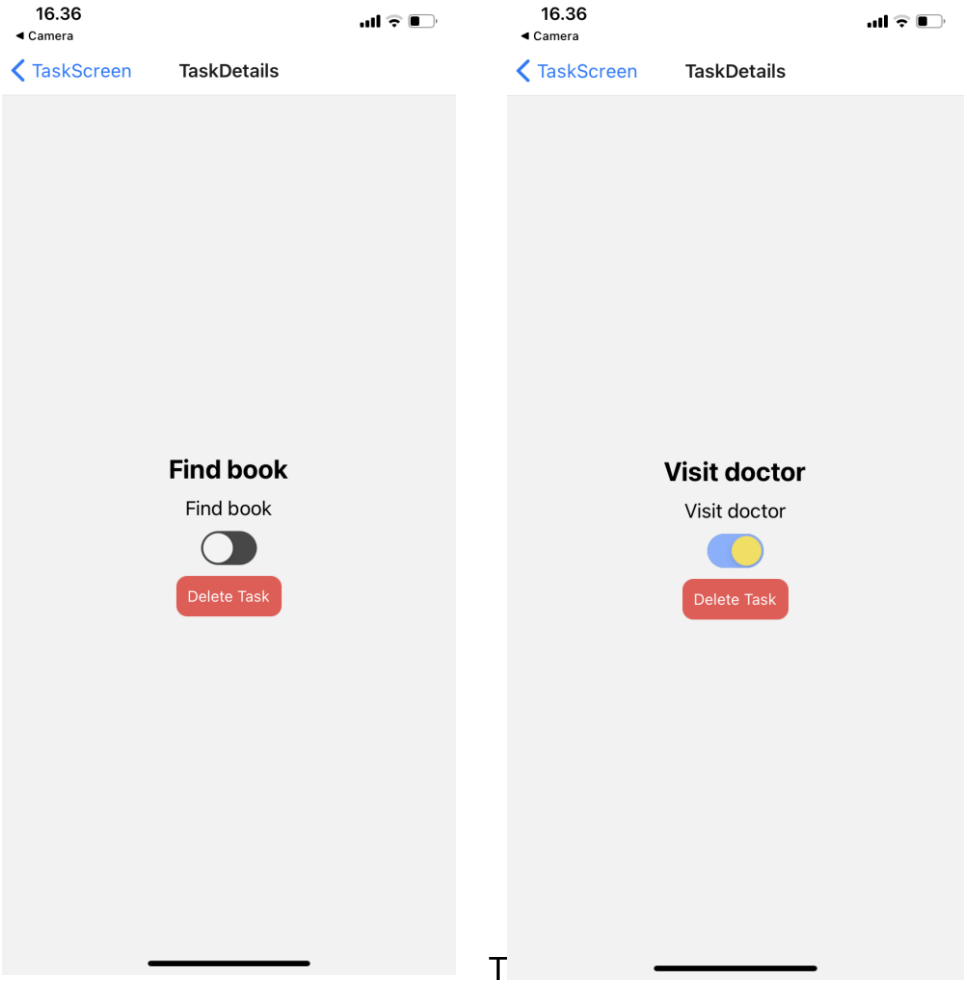
```
const [filterStatus, setFilterStatus] = useState('all');
const [filteredTasks, setFilteredTasks] = useState<FirestoreTask[]>(tasks);

useEffect(() => {
  if (filterStatus === 'all') {
    setFilteredTasks(tasks);
  } else if (filterStatus === 'completed') {
    setFilteredTasks(tasks.filter(task => task.task.completed));
  } else if (filterStatus === 'not completed') {
    setFilteredTasks(tasks.filter(task => !task.task.completed));
  }
}, [tasks, filterStatus]);
```

Figure 16. React Hook

3.4.4 Task Details

The Task Details screen (Figure 17) provides users with detailed information about a selected task, such as the task name and description. Users can also mark the task as completed and delete the task from the list. The screen includes button for deleting the task.



T

Figure 17. Task Details screen with examples of completed and uncompleted task.

The task deletion feature has been implemented with Firestore, allowing users to delete tasks from their task list with ease.

There is a Switch component in Task Detail screen, which is used to toggle the completion status of a task (Figure 18). The Switch component is a built-in React Native component that provides a visual toggle control. The “trackColor” prop is used to define the colours of the switch when it is in the 'off' and 'on' positions. The “thumbColor” prop is used to define the colour of the switch toggle. In this case, the “thumbColor” changes based on whether the task is completed or not.

```
const [completed, setCompleted] = useState(task.task.completed);

const handleToggleCompleted = (value: boolean) => {
  const updatedTask = { ...task, completed: value };
  onUpdateTask(updatedTask);
  setCompleted(value);
};

  return (
    <View style={styles.container}>
    <Text style={styles.title}>{task.task.title}</Text>
    <Text style={styles.description}>{task.task.description}</Text>
    <Switch       You, 2 weeks ago • Add different components
        style={styles.completedToggle}
        trackColor={{ false: '#767577', true: '#81b0ff' }}
        thumbColor={completed ? '#f5dd4b' : '#f4f3f4'}
        ios_backgroundColor="#3e3e3e"
        value={completed}
        onValueChange={handleToggleCompleted}
      />
```

Figure 18. Switch component.

4 Results

4.1 Advantages of Using ChatGPT for Development

AI language models such as ChatGPT are becoming increasingly popular in the world of software and mobile app development as they can provide developers with a wide range of benefits.

ChatGPT can help mobile app developers improve their workflows. With an AI language model, you can automate tasks, optimize coding practices, and identify potential areas for improvement. AI language models can also provide language-specific recommendations and fixes to ensure code is accurate and functional.

One of the main advantages of using ChatGPT in mobile application development is its ability to understand and respond to natural language input. This means that users can interact with the application in a more natural and intuitive

way, which can lead to a more engaging and enjoyable user experience. With ChatGPT, developers can create chatbots, voice assistants, or other NLP features for their mobile applications that can understand user requests in natural language and respond in a useful way. (Top benefits of ChatGPT in Mobile application development, 2023).

Another important benefit is the ability to automate repetitive tasks. This can help developers save time and focus on more complex and creative tasks such as developing new features and improving the user experience. For example, AI language models can automatically generate snippets of code based on user input, reducing the time and effort required to write new code from scratch.

The chatbot can identify potential areas for optimization in the code. By analysing large amounts of code, AI language models can help identify inefficiencies or areas where performance can be improved. This can help developers optimize their code and ensure it runs as efficiently as possible.

Additionally, AI language models can provide language-specific recommendations and corrections to ensure code accuracy and functionality. This can help developers catch errors and improve code quality, leading to more stable and reliable software applications.

ChatGPT has had a significant impact on mobile app development. Developers are using this technology to build better and smarter chatbots that can understand and respond to complex queries, provide personalized recommendations, improve user engagement, and enhance customer support. (Saleem 2023).

4.2 Limitations of Using ChatGPT for Development

Despite the many improvements that Chat-GPT has made, it still has certain limitations. One major limitation is that it cannot yet adapt to changes in conversation or context. Although it understands very well what the user is saying and generates an appropriate response, it does not always help to understand the

changing context of the conversation. This may limit the success of the machine conversation and may mean that the conversation may get stuck or cannot proceed any further. (Bansal 2023).

The chatbot can only provide answers from some time ago. ChatGPT data ends in 2021. This can be a problem for those who don't know and can result in incorrect information being added when writing an essay or posting on a blog. The fact is that AI can be useful when the user continues to use it with the utmost caution. (The Limitations of ChatGPT 2023).

Sometimes ChatGPT gives incorrect answers, which may be due to insufficient training. Stackoverflow, a coding website, has officially banned the output generated by chatbot as most of it was incorrect. ChatGPT often provides answers that appear to be correct, even though they may be incorrect. Anyone who is not an expert can take them for the truth. Even if the question is not particularly difficult, ChatGPT can still give a very wrong answer. (The Limitations of ChatGPT 2023).

Limited domain expertise. Domain expertise refers to the knowledge and experience in a particular field or industry. ChatGPT is a general-purpose language model and may not have the same level of expertise as a human expert in a specific domain. Developers can fine-tune ChatGPT on domain-specific data to improve its domain expertise, but it requires a significant amount of data and expertise.

4.3 Best Practices for Utilizing ChatGPT

The interesting thing about the ChatGPT is that you can have an ongoing conversation with it, and it will remember your previous interactions.

As an example, when I requested ChatGPT to assist me with integrating Firebase Authentication to my project without providing any additional details (Figure 19), it was able to recall from previous prompts that I was working on a

React Native application using TypeScript, and as a result, it provided me with code that was adapted to the correct language and framework.

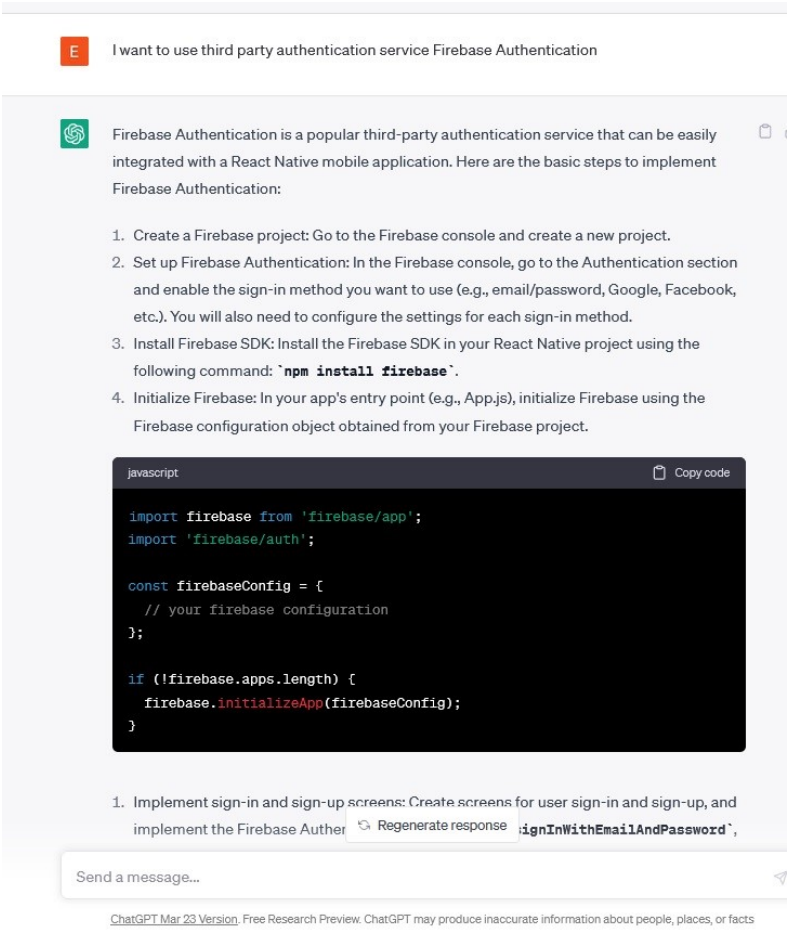


Figure 19. Prompt Example.

To ensure that ChatGPT responds accurately and effectively, it is important to provide it with clear and detailed prompts. For example, unclear prompt: "Help me with my app", in contrast clear prompt: "I'm developing a mobile task management application using React Native and Firebase. I need assistance with implementing the Sign In and Sign Out screens, as well as creating a Firestore collection for storing task data. Can you provide guidance on these tasks and suggest best practices for improving the user experience of the application?"

If you want ChatGPT to assist you with adding a new feature to your application, it is important to provide specific details such as the name of the feature,

the functionality it should have, and any relevant constraints or limitations. Additionally, you could provide examples of similar features in other applications to give ChatGPT a better understanding of your requirements. This will help ensure that the suggestions and advice provided by ChatGPT are relevant and useful to your specific needs.

Another way to use ChatGPT as a developer helper when building a mobile application is to integrate it into your Integrated Development Environment (IDE). Many popular IDEs, such as Visual Studio Code, offer extensions that allow you to use ChatGPT directly within your development environment.

Here is an example, how Code GPT extension can be used within Visual Studio Code (Figure 18). It is possible to generate code using ChatGPT, simply add a comment (here is “create typescript function to get completed tasks”) for the task you wish to accomplish and press the “Ctrl + Shift + I” keys together.

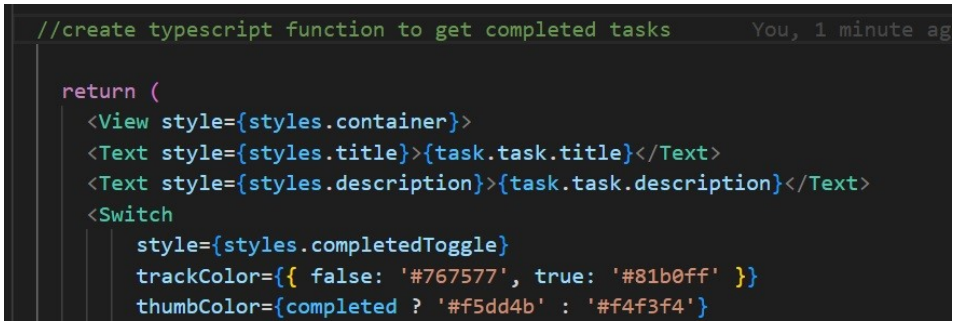


Figure 20. Comment for the task you want accomplish.

This will generate the code on the right side (Figure 21).

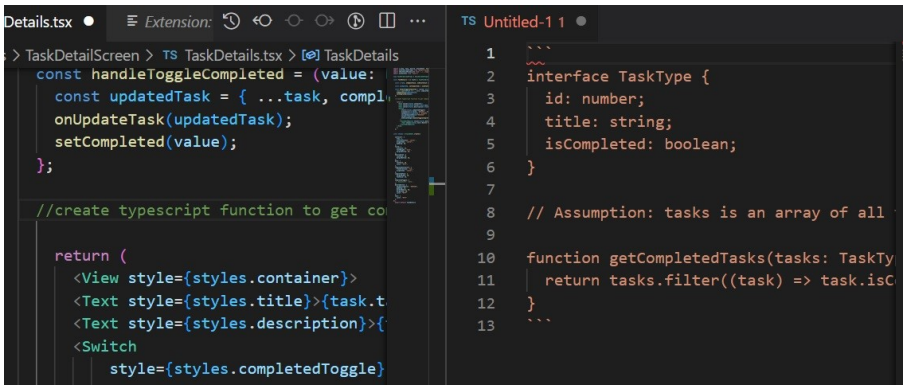


Figure 21. Code generated by ChatGPT.

Apart from that, it is possible to select a code, right-click on it, and ask Code GPT to explain the code, refactor it, find problems, debug, and more (Figure 22).

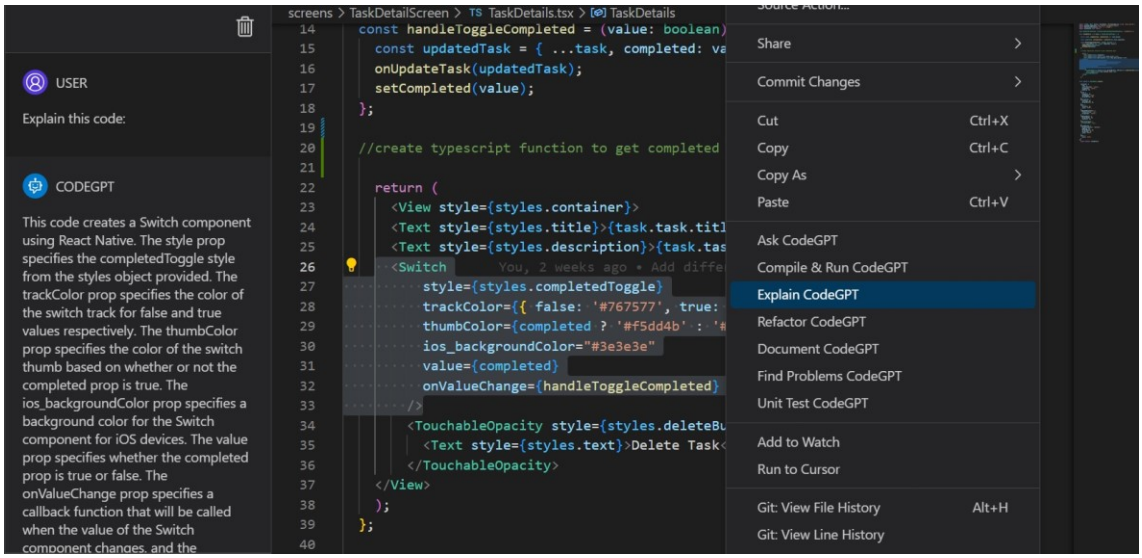


Figure 20. How to utilize Code GPT within Visual Studio Code.

5 Conclusion

The purpose of the engineering work was to develop a mobile application for the task management using React Native technology, Firebase Authentication and Firestore with the ChatGPT assistance and at the same time to study the

use of AI language model in the programmer's work, to examine the language models used to create chatbots and to study the basic concepts of ChatGPT.

In the research phase, various materials, and articles about ChatGPT were explored to gain a comprehensive understanding of language models and artificial intelligence. The history of ChatGPT and its creators were also studied, providing valuable insights into its capabilities and potential applications.

Artificial intelligence is developing more and more every day, and ChatGPT is a clear proof of this. This versatile chatbot can already compete with humans in intellectual and communicative tasks. The use of ChatGPT is expanding, and this trend will continue.

The process of Task Management application development went relatively smoothly due to the previous familiarity with React Native technologies and Firebase fundamentals. ChatGPT was a helpful tool during development, providing ready-made code snippets. However, there were also various errors encountered that ChatGPT was unable to assist with resolving. In general, ChatGPT has the potential to improve developer’s workflow and increase efficiency but requires some field knowledge to be fully effective.

ChatGPT is a good programmer but not a good software designer or developer. This AI cannot design software from an engineer's perspective with business needs in mind or fix errors other than those detected in its training data. At least not for now. This tool cannot replace those whose work goes beyond routine coding. On the contrary, this tool can open a wide range of possibilities for those who work with the code. Just need to give the right instructions to this AI language assistant and it can make the workflow easier. This leads us to the following conclusion: ChatGPT is the beginning of more powerful developers.

References

A Guide to OpenAI and its Products and Services. 2023. Online. The Windows Club. <https://www.thewindowsclub.com/a-guide-to-openai-and-its-products-and-services>. 9.1.2023. Accessed 7.2.2023.

Bansal, Yash. 2023. Limitations of Chat-GPT. Online. Medium. <https://medium.com/@yashbansal42/the-limitations-of-chat-gpt-and-the-need-for-further-research-and-development-91bc6caae533>. 29.3.2023. Accessed 25.4.2023.

Bard, Jonah. 2021. How GPT-3 Actually Works, From the Ground Up. Online. Medium. <https://medium.com/nerd-for-tech/how-gpt-3-actually-works-from-the-ground-up-5714ae7f3355>. 9.3.2021. Accessed 25.2.2023.

Borozenets, Michael. 2022. React Native Init vs Expo 2022: What Are the Differences? Online. Fulkrum. < https://fulcrum.rocks/blog/react-native-init-vs-expo#what-is-expo>. 22.4.2022. Accessed 14.2.2023.

Cooper, Kindra. 2021. OpenAI GPT-3: Everything You Need to Know. Online. Springboard. <https://www.springboard.com/blog/data-science/machine-learning-gpt-3-open-ai/>. 1.11.2021. Accessed 8.2.2023.

Cloud Firestore. 2023. Online. Firebase Documentation. <https://firebase.google.com/products/firestore>. 13.4.2023. Accessed 17.4.2023.

Cuofano, Gennaro. 2023. Instruct GPT And Why It Matters For The Success Of ChatGPT. Online. FourWeekMBA. <https://fourweekmba.com/instructgpt/>. 13.2.2023. Accessed 25.2.2023.

Dongre, Anay. 2023. Discover the Revolutionary Instruct GPT. Online. OpenGenius IQ. <https://iq.opengenus.org/instruct-gpt/>. 2023. Accessed 25.2.2023.

Eisenman, Bennie. 2015. Learning React Native. E-book. O'Reilly.

Firebase Authentication. 2023. Online. Firebase Documentation. <https://firebase.google.com/docs/auth>. 13.4.2023. Accessed 17.4.2023.

Frankenfield, Jake. 2022. Artificial Intelligence: What It Is and How It Is Used. Online. Investopedia. <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>. 6.7.2022. Accessed 7.2.2023.

History of Artificial Intelligence. 2020. Online. Council of Europe. <https://www.coe.int/en/web/artificial-intelligence/history-of-ai>. 6.7.2020. Accessed 7.2.2023.

Kapronczay, Mór. 2022. A Beginner's Guide to Language Models. Online. BuiltIn. <https://builtin.com/data-science/beginners-guide-language-models>. 13.12.2022. Accessed 7.2.2023.

Lowe, Ryan & Wainwright, Caroll. 2022. Instruct GPT Model Card. Online. GitHub. <https://github.com/openai/following-instructions-human-feedback/blob/main/model-card.md>. 1.2022. Accessed 25.2.2023.

Maldonado, Leonardo. 2019. A Short Introduction to TypeScript. Online. Progress Telerik. <https://www.telerik.com/blogs/a-short-introduction-to-typescript>. 26.8.2019. Accessed 17.4.2023.

Ot, Annina. 2022. What Is OpenAI and Does It Really Make Coding Easier? Online. MUO. <https://www.makeuseof.com/openai-coding/>. 19.1.2022. Accessed 7.2.2023.

Pocock, Gennaro. 2023. What Is ChatGPT? – what is it used for? Online. PCGuide. <https://www.pcguide.com/apps/what-is-chat-gpt/>. 13.2.2023. Accessed 25.2.2023.

Reyes, Kate. 2023. What is Deep Learning and How Does It Works. Online. SimpliLearn. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning/>. 17.2.2023. Accessed 7.5.2023.

Roldós, Inés. 2020. NLP, Machine Learning & AI, Explained. Online. MonkeyLearn. <https://monkeylearn.com/blog/nlp-ai/>. 9.6.2020. Accessed 25.2.2023.

Saleem, Aaliya. How ChatGPT Impact On Mobile App Development. 2023. Online. Medium. < https://mobileappcircular.com/how-chat-gpt-impact-on-mobile-app-development-d5bdc37b0513>. 15.3.2023 Accessed 25.4.2023.

The Limitations of ChatGPT. 2023. Online. Nandbox. <https://nandbox.com/the-limitations-of-chatgpt/>. 23.1.2023 Accessed 25.4.2023.

Top benefits of ChatGPT in Mobile application development. 2023. Online. Sieg Partners. <https://www.siegpartners.com/top-benefits-of-chatgpt-in-mobile-application-development/>. Accessed 25.4.2023.

Wodeski, Ben. 2022. 7 language models you need to know. Online. AI Business. <https://aibusiness.com/nlp/7-language-models-you-need-to-know>. 27.7.2022. Accessed 25.2.2023.

What is GPT-3, How Does It Work, and What Does It Actually Do? 2021. Online. Medium. <https://medium.com/sciforce/what-is-gpt-3-how-does-it-work-and-what-does-it-actually-do-9f721d69e5c1>. 17.9.2021 Accessed 7.2.2023.