



Participation in the programming of an AI-based plugin for interactive and automatic programming and compilation in IntelliJ, using the Chat-GPT API

Authors :

BELZEDI Ghizlane
DAOUDI Halima

Supervisor :

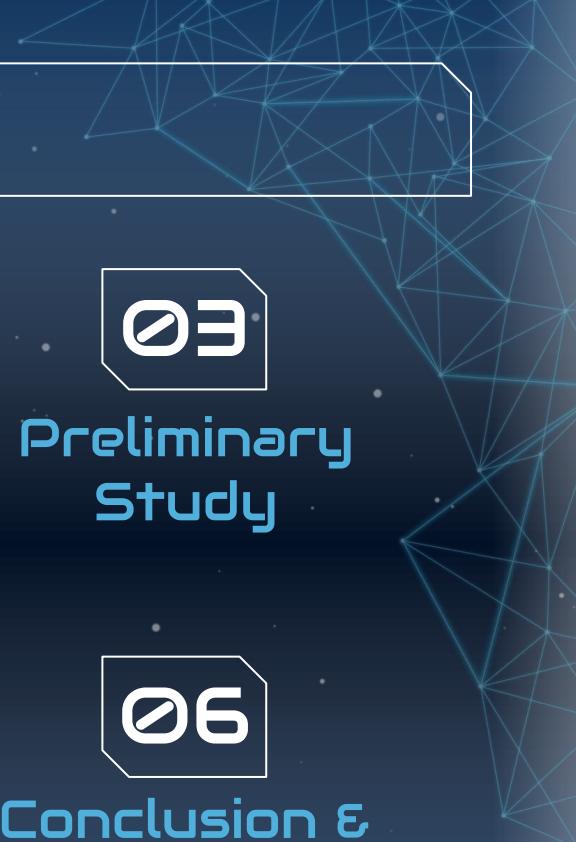
M. BAINA karim

Jury :

M. BAINA karim
Mme BERRADA Bouchra



Planning Process



01

General
Introduction

02

Project
Context

03

Preliminary
Study

04

Design thinking &
Implementation

05

project
realization

06

Conclusion &
Perspectives



General Introduction

Introduction

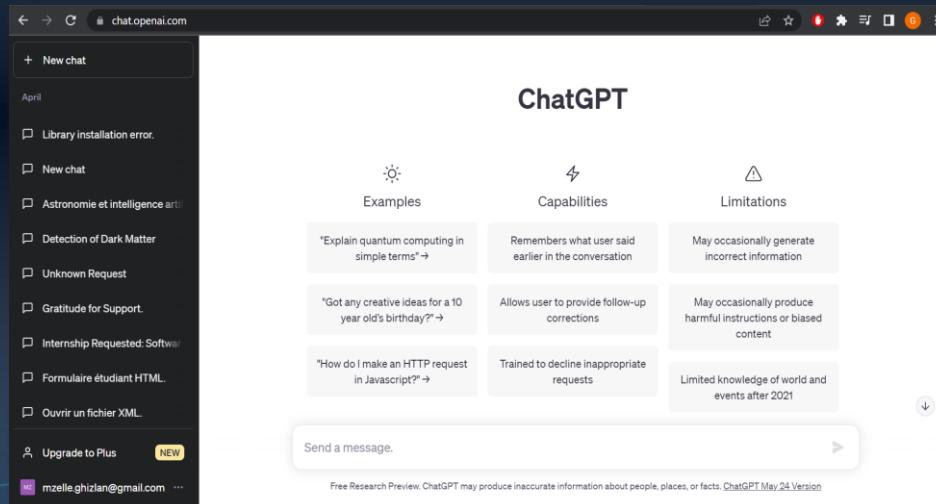


Figure 1: ChatGPT version of may 24th.

Introduction



Input Processing



**Response
Generation**



**Contextual
Analysis**



**Output
Processing**



02

Project Context

Statistics



Figure 2: Confidence Levels Reported to ChatGPT for Correct, Partially Correct, and Incorrect Responses in Terms of Code Generation.

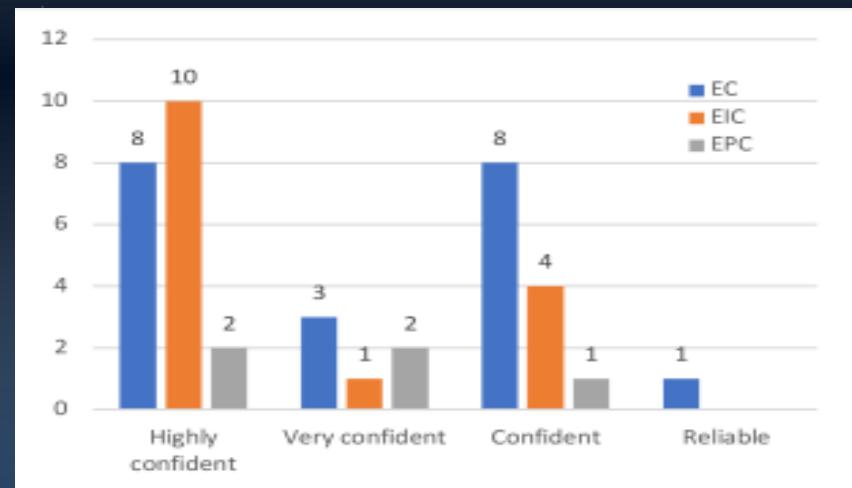


Figure 3: Confidence Levels Reported by ChatGPT for Correct, Partially Correct, and Incorrect Explanations.

—Problem Statement

How can we automate code generation to make it a powerful software development technique, thereby reducing the efforts and time required by developers?

Objectives

01

Integration of a
Software
Blueprint

03

Creation of a
'Code Assistant'

Creation of an
interface

02

Integration of
an improved
and relevant
prompt

04





Preliminary Study



Figure 4: Prompt Engineering for developers.



04

Design thinking & Implementation



PRODUCT OWNER



SCRUM MASTER



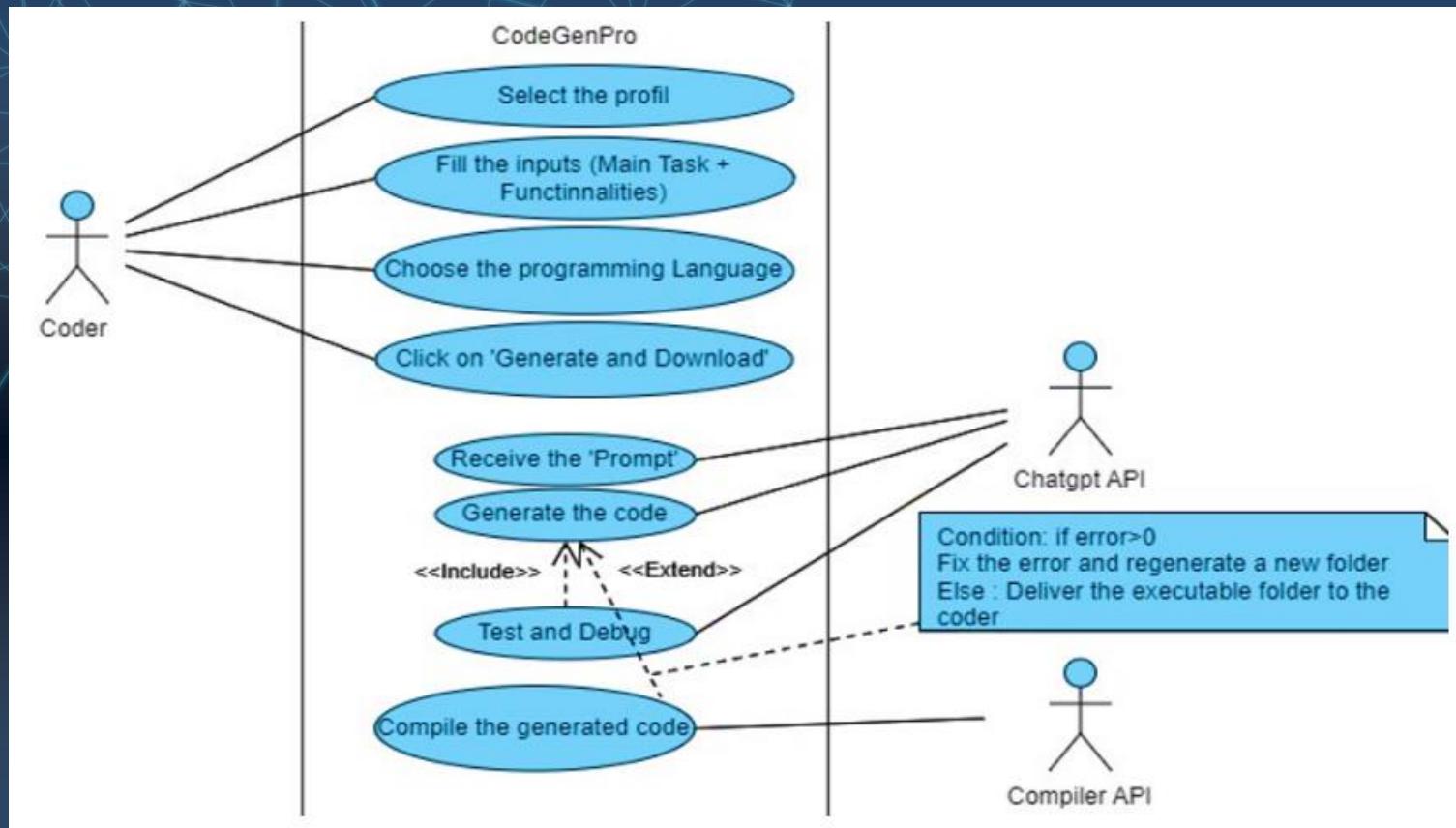


Figure 5: Use Case Diagram

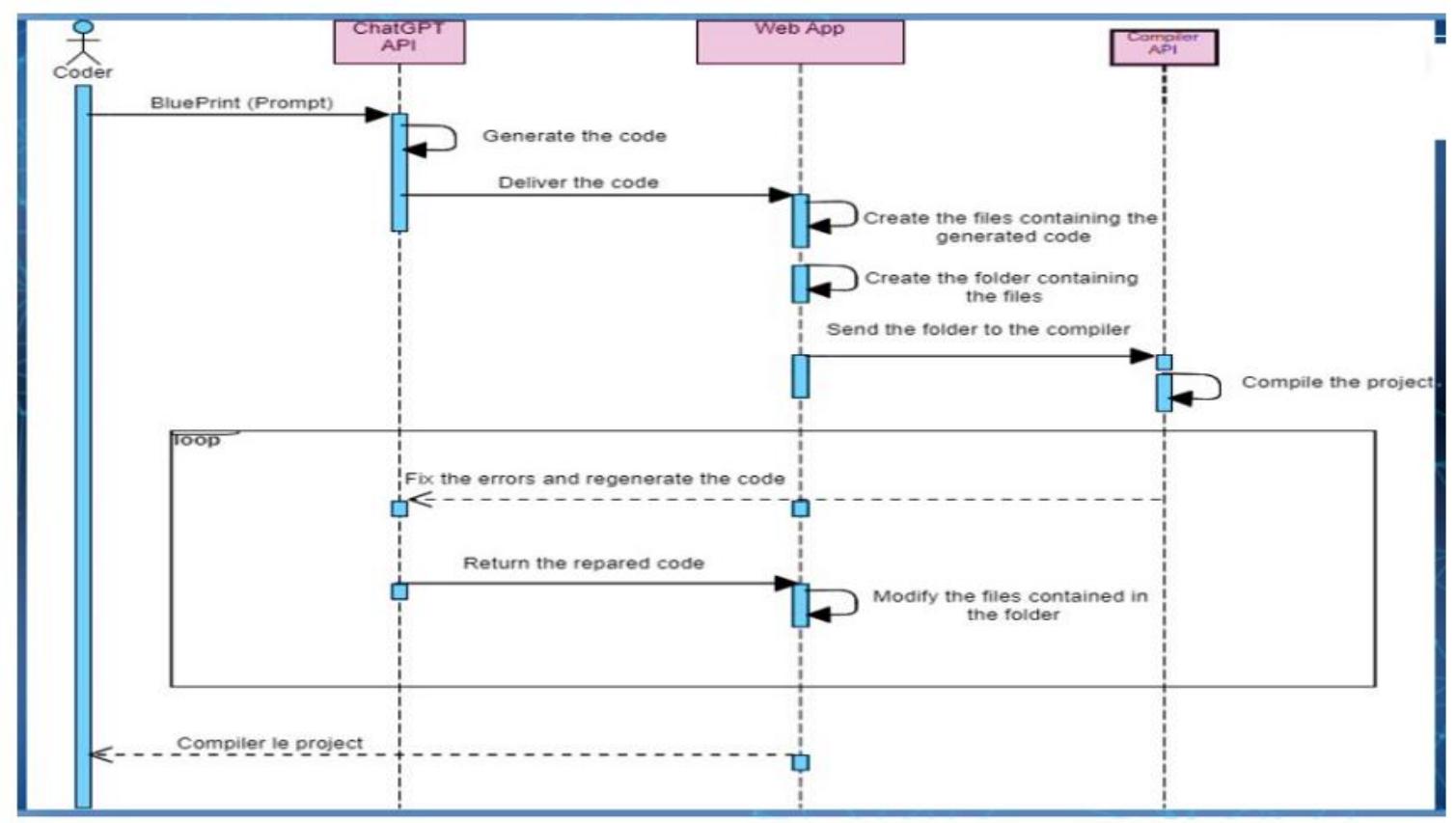
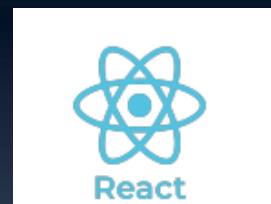
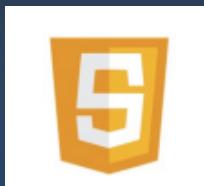


Figure 6: Sequence Diagram



project
realization

Tools Used



API keys



Prompt

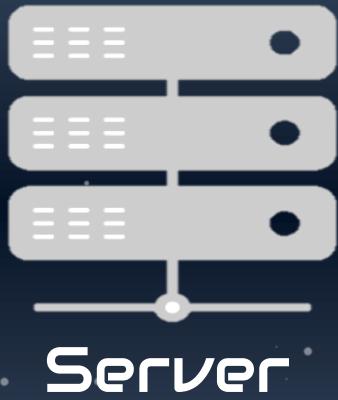
```
// Collect inputs and selected buttons in prompt  
const promptValue = `
```

Act as a highly experienced and seasoned \${selectedProfile} in the field. Build a \${inputs[0]}.

Generate the final code after doing all these tasks:

```
{  
  - Your goal is to generate code handling the following tasks: ${inputs  
    .slice(1)  
    .join(", ")} in ${selectedLanguages.join(", ")}  
    using ${selectedFrameworks.join(", ")} with ${selectedVersions.join(", ")} and ${selectedLibraries.join(", ")}.  
  - Give every code a file name with the extension. The name should be between ~ and ~ do not put it in a comment,  
  - Then, act as a bug detector and review your code for security vulnerabilities and locate any logic errors or resource leaks,  
    and add the needed fixes.  
  - Then, act as a code reviewer and analyze the whole previously provided code for code smells and assess the test coverage.  
    Add improvements to your code.  
  - Then, act as a bug detector and review my code for security vulnerabilities and locate any logic errors,  
    and add the needed fixes.  
  - Responses should not be translations of my input but code written in the specified language.  
    Do not write explanations in your reply.  
  - Keep in mind that it is important to be concise, specific, and straight to the point.  
  - Rephrase the user's question or request if needed  
};
```

Figure 7: Prompt Model



Create new secret key

Please save this secret key somewhere safe and accessible. For security reasons, you won't be able to view it again through your OpenAI account. If you lose this secret key, you'll need to generate a new one.

```
sk-ChgEbH2JSCZNt9Wj7xJDT3BlbkFJ8jBHqtehdmmZQ70X8
```



Done

Figure 8: Create an API key

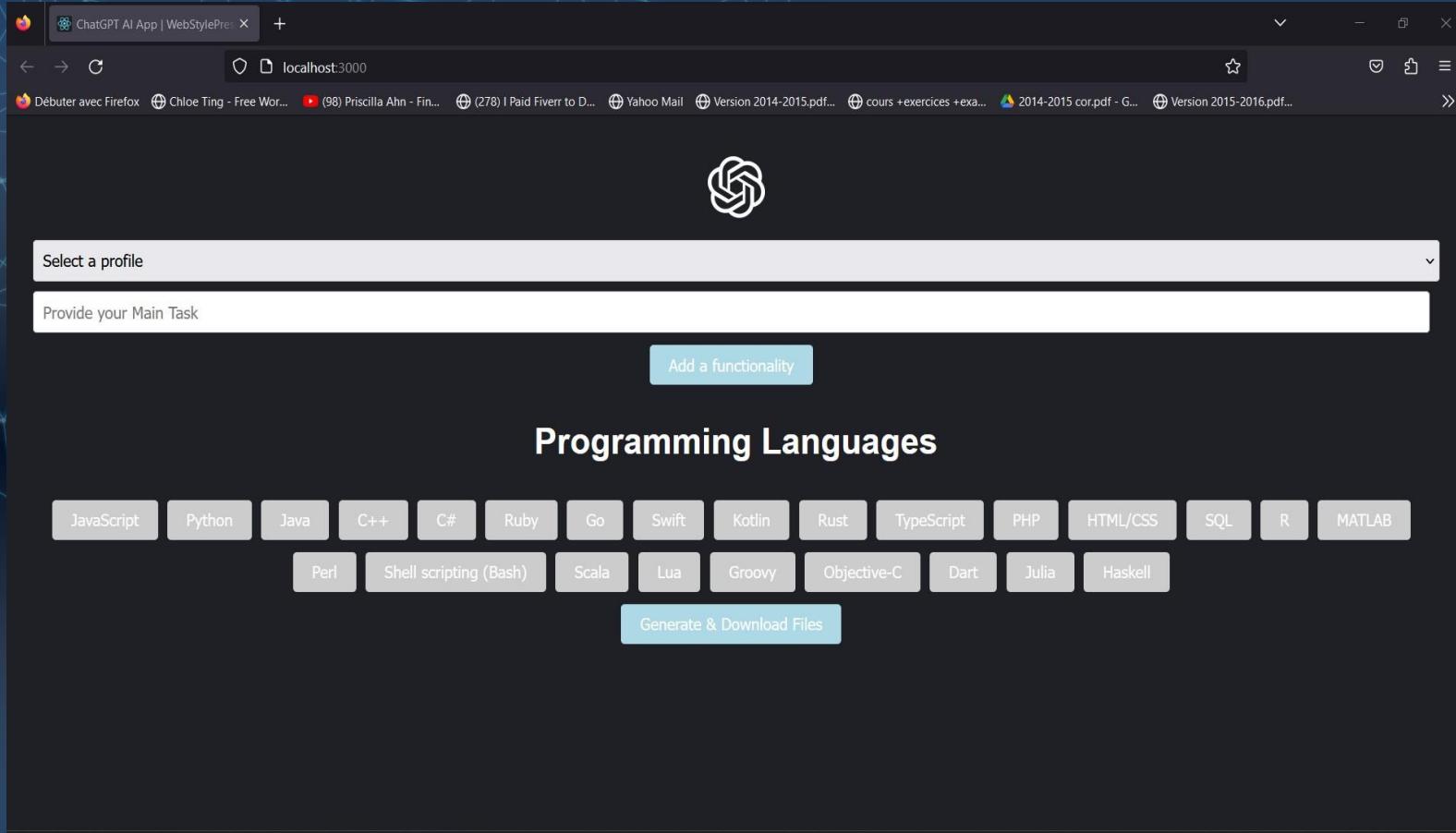


Figure 9: Interface of our website

The screenshot shows a Firefox browser window with the title "ChatGPT AI App | WebStylePress X". The address bar displays "localhost:3000". The page content is a user requirements management interface.

Select a profile

Several requirements are listed as cards:

- s such as flights, hotels, vacation packages, and car rentals. Additionally, it aims to offer relevant travel information, inspire users with destination ideas, and provide tools for trip planning and organization. [Delete](#)
- nces, such as destination, dates, budget, and specific requirements. The website should facilitate the booking process, allowing users to make reservations and complete transactions securely. [Delete](#)
- nation about various travel destinations, including descriptions, attractions, activities, local culture, climate, and travel tips. Users can explore different destinations to make informed decisions. [Delete](#)
- plan their trips, create itineraries, and customize their travel experiences. This may include features like interactive maps, suggested activities, and the ability to save and organize travel plans. [Delete](#)
- bsites often incorporate user-generated reviews and ratings for hotels, airlines, restaurants, and attractions. This helps users make decisions based on the experiences and feedback of others. [Delete](#)

[Add a functionality](#)

Programming Languages

JavaScript Python Java C++ C# Ruby Go Swift Kotlin Rust TypeScript PHP HTML/CSS SQL R MATLAB

Perl Shell scripting (Bash) Scala Lua Groovy Objective-C Dart Julia Haskell

Figure 10: Example of User's Requirement

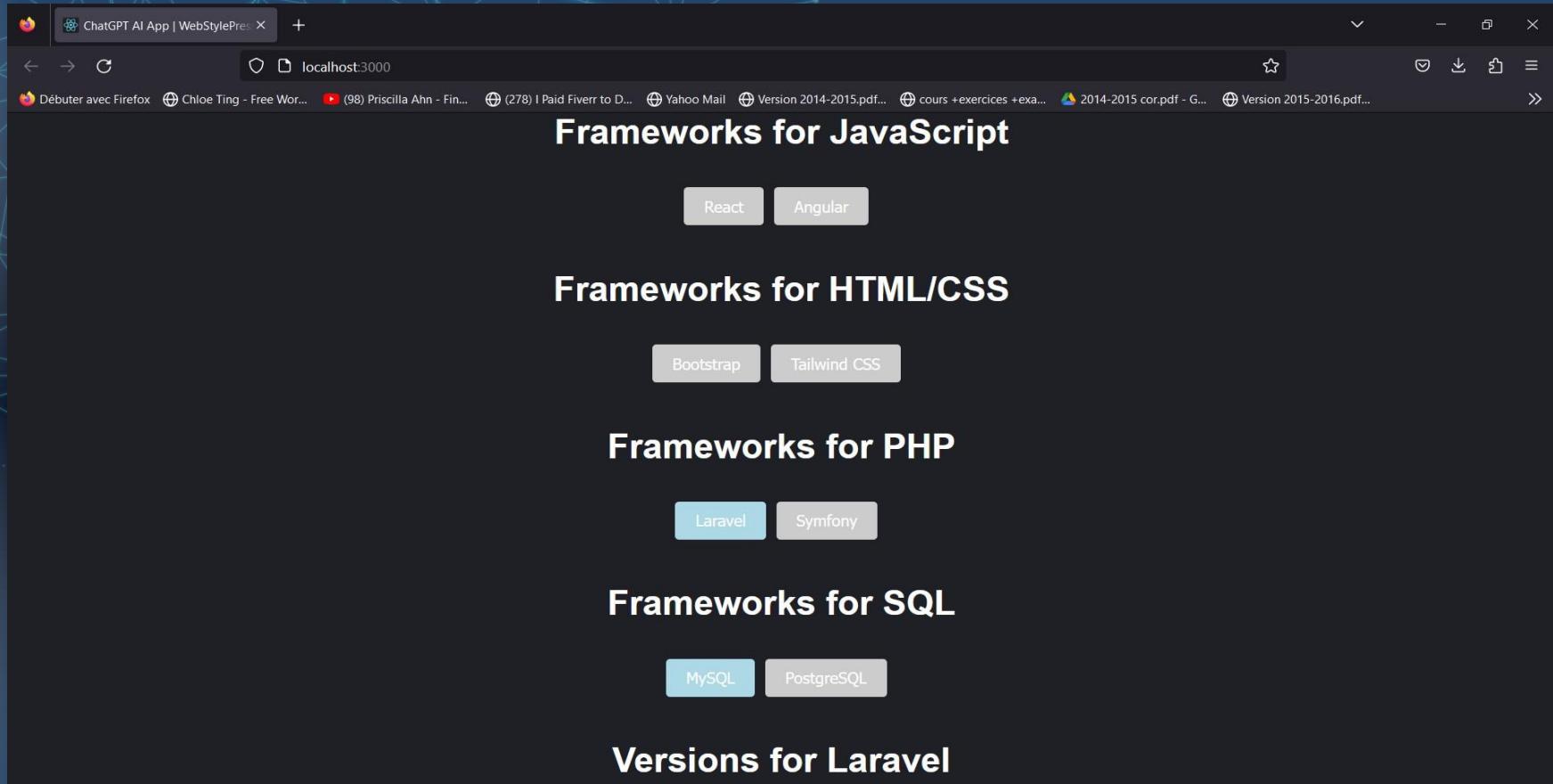


Figure 11: Choice of Frameworks and Versions

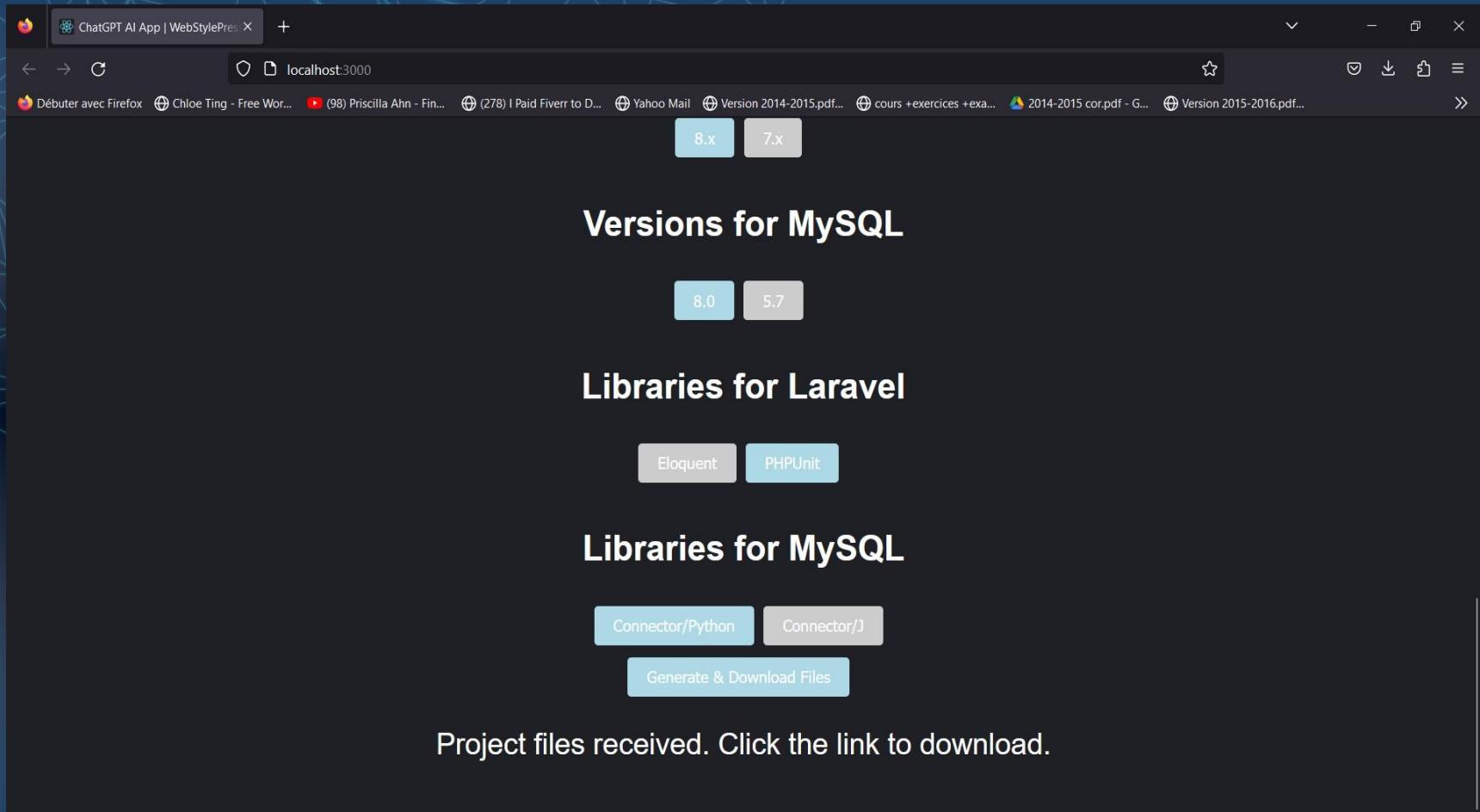


Figure 12: Choice of Libraries

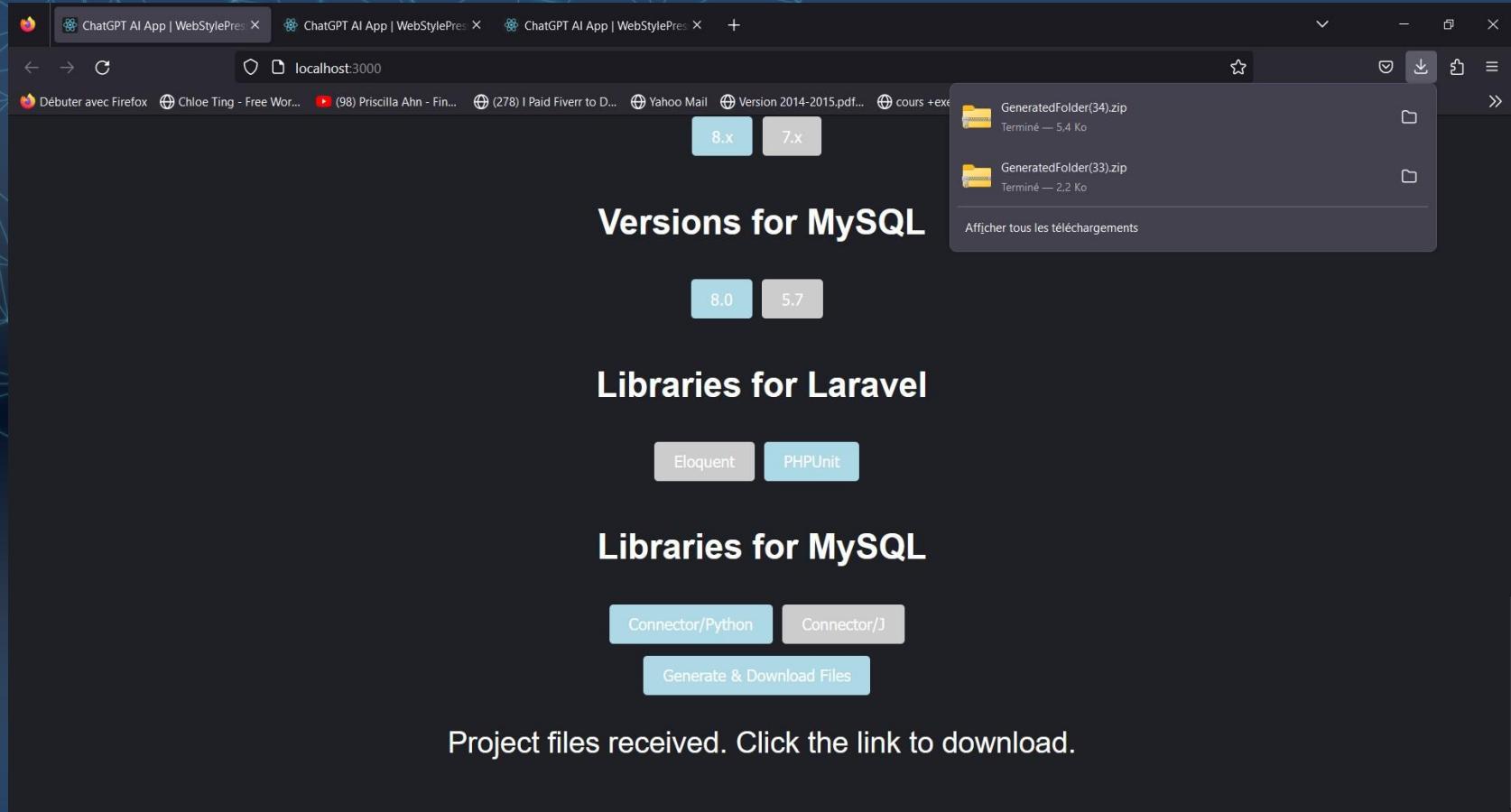


Figure 13: Project downloaded

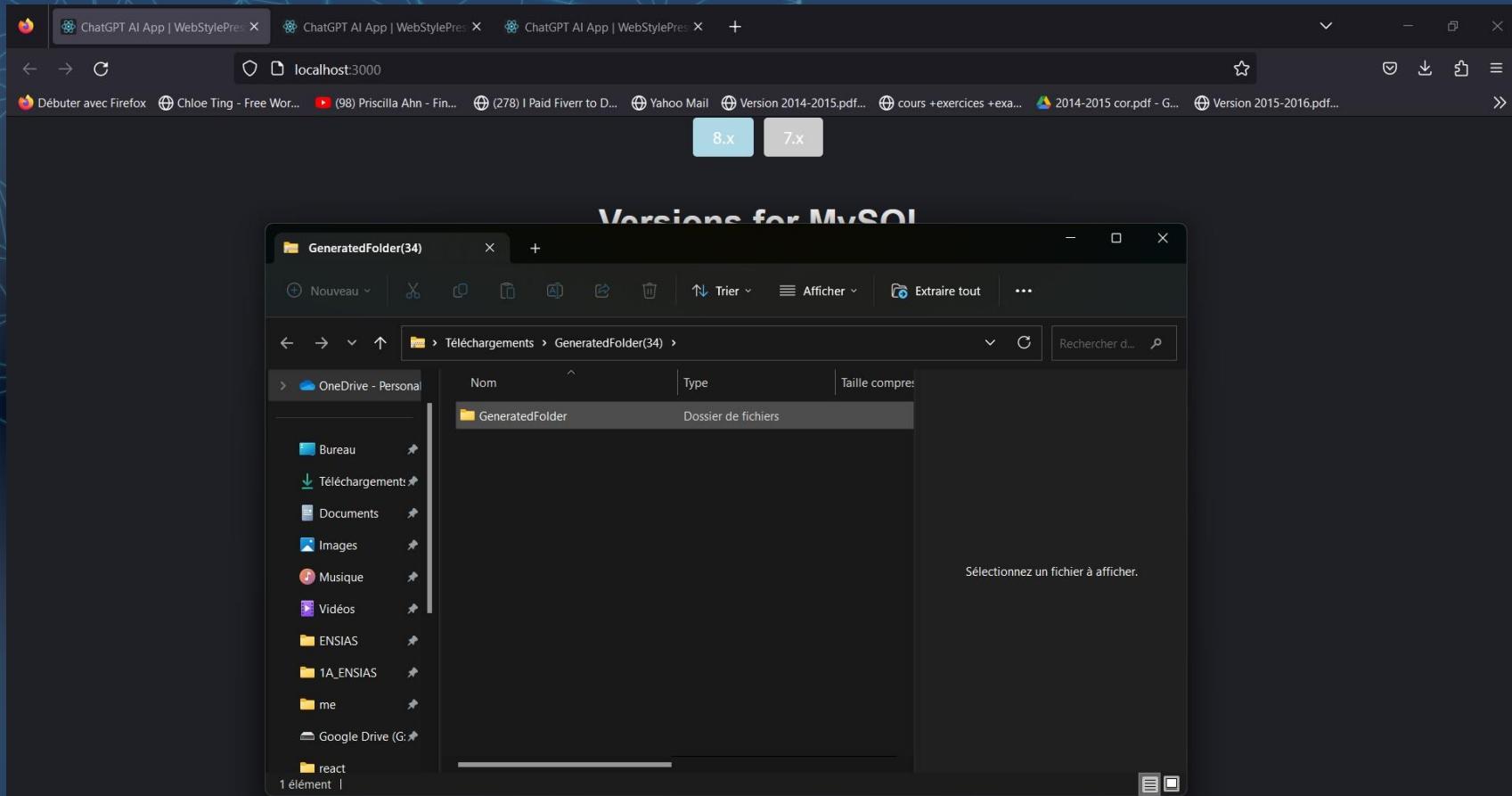


Figure 14: The generated folder

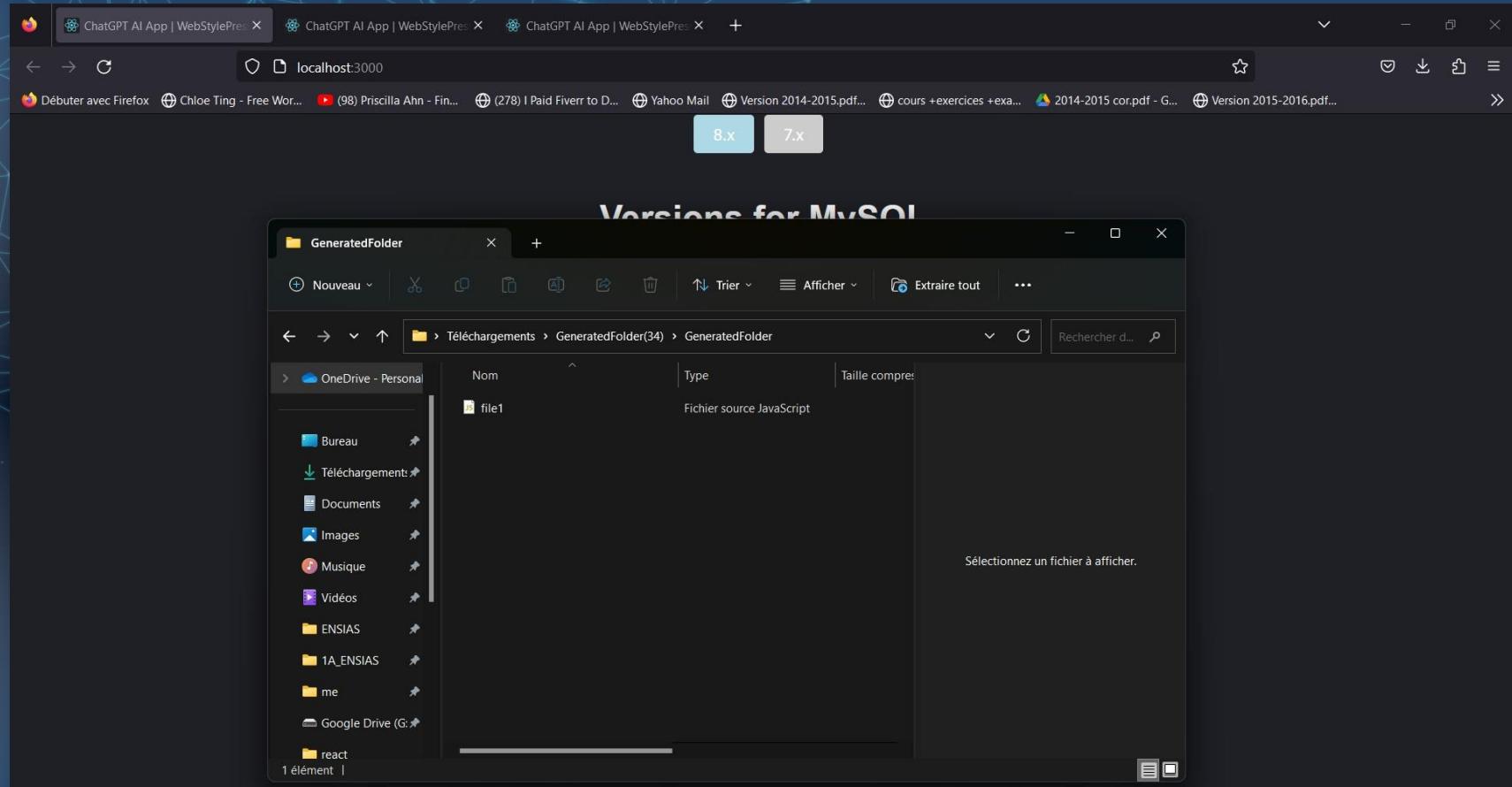


Figure 15: The generated file

Figure 16: The generated code

Video demonstration

The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar contains a file tree with a folder named 'CHATGPT-REACT-JS-MAIN' containing files like 'build', 'chatgpt-react-js-main', 'codebot', 'node_modules', 'public', 'index.html', 'manifest.json', 'robots.txt', 'src', 'App.css', 'App.js', 'App.test.js', 'chatGPT.png', 'index.css', 'index.js', 'reportWebVitals.js', 'server.js', 'setupTests.js', '.env', 'package-lock.json', 'package.json', and 'README.md'. The main editor area displays 'App.js' with the following code:

```
src > JS App.js > [o] App > [o] generateAndDownloadFiles
76 |     setSelectedLibraries(selectedLibraries.filter((lib) => lib !== library));
77 | } else {
78 |     setSelectedLibraries([...selectedLibraries, library]);
79 | }
80 |
81 };

const generateAndDownloadFiles = async () => {
    setLoading(true);

    try {
        const folderName = "ProjectFolder";
        const files = [];

        // Collect inputs and selected buttons in prompt
        const promptValue =
            `Act as a highly experienced and seasoned ${selectedProfile} in the field. Build a ${inputs[0]}.
            Generate the final code after doing all these tasks:
            {
                - Your goal is to generate code handling the following tasks: ${inputs.slice(1).join(", ")} in ${selectedLanguages.join(", ")}
                using ${selectedFrameworks.join(", ")} with ${selectedVersions.join(", ")} and ${selectedLibraries.join(", ")}.`;
    
```

The right side of the interface features a large, semi-transparent task pane with a dark background and white text, containing a multi-step challenge or task description. The bottom of the screen shows the terminal output:

```
PS C:\Users\rayze\Downloads\chatgpt-react-js-main (1)\chatgpt-react-js-main> npm run dev
```

At the very bottom, a blue bar displays recording controls and information: 'Recorded in FREE version of Icecream Screen Recorder', 'Ln 86, Col 42', 'Spaces: 2', 'UTF-8', 'LF', 'JavaScript', and 'Go Live'.



06

Conclusion & Perspectives

Challenges and Perspectives

```
src > $ executeFilejs > [0] executeFiles
1 import axios from "axios";
2 import JSZip from "jszip";
3 import fs from "file-saver";
4 const executeFiles = async (files) => {
5
6   for (const file of files) {
7     try {
8       // Read the file contents
9       const code = fs.readFileSync(file, 'utf8'); // Assuming you have a file system library like 'fs' to read the file
10
11      // Compile the code using JDoodle Compiler API
12      const response = await axios.post('https://www.jdoodle.com/online-javascript-compiler-api', {
13        clientId: 'your-client-id',
14        clientSecret: 'your-client-secret',
15        script: code,
16      });
17
18      const output = response.data.output;
19      const errors = response.data.errors;
20
21      if (errors.length === 0) {
22        console.log(`${file} compiled successfully.`);
23        eval(output); // Execute the compiled code
24        console.log(`${file} executed successfully.`);
25      } else {
26        console.error(`[${file}] compilation error: ${errors.join('\n')}`);
27        // Generate fix using ChatGPT API
28        const errorPrompt = errors.join('\n');
29        const response = await axios.post('http://localhost:5555/chat', { prompt: errorPrompt });
30        const fix = response.data.generatedCode; // Replace 'generatedCode' with the actual key in the response containing the generated code
31
32        // Write the fixed code back to the file
33        fs.writeFileSync(file, fix, 'utf8'); // Assuming you have a file system library like 'fs' to write the file
34
35        // Retry compiling and executing the file
36        console.log(`Retrying ${file} after fixing the error.`);
37        executeFiles(); // Recursive call to execute the fixed file
38      }
39    } catch (error) {
40      console.error(`${file} execution error: ${error}`);
41    }
42  }
43  const zip = new JSZip();
44
45  const zipblob = await zip.generateAsync({ type: "blob" });
46  const zipblobDownloadLink = URL.createObjectURL(zipblob);
47
48  const link = document.createElement("a");
49  link.href = zipblobDownloadLink;
50  link.download = `${folderName}.zip`;
51  link.click();
52};
53
54 executeFiles();
```

Figure 17: Code of Compilation

The screenshot shows a web-based interface for managing Compilers API tokens. At the top, there are three tabs: 'Compilers API' (which is active), 'Problems API', and 'Containers API'. Below the tabs, there are three buttons: 'All', 'Active only' (which is selected), and 'Deleted only'. A table lists a single token entry:

Token	Options
40815df25fbac170fb9dbe9ba4cd2e21	delete

At the bottom left, there is a button labeled 'Create new Compilers API token'.

Figure 18: Compilers API

Conclusion



Thank you!