TEST CASE GENERATOR

INTRODUCTION

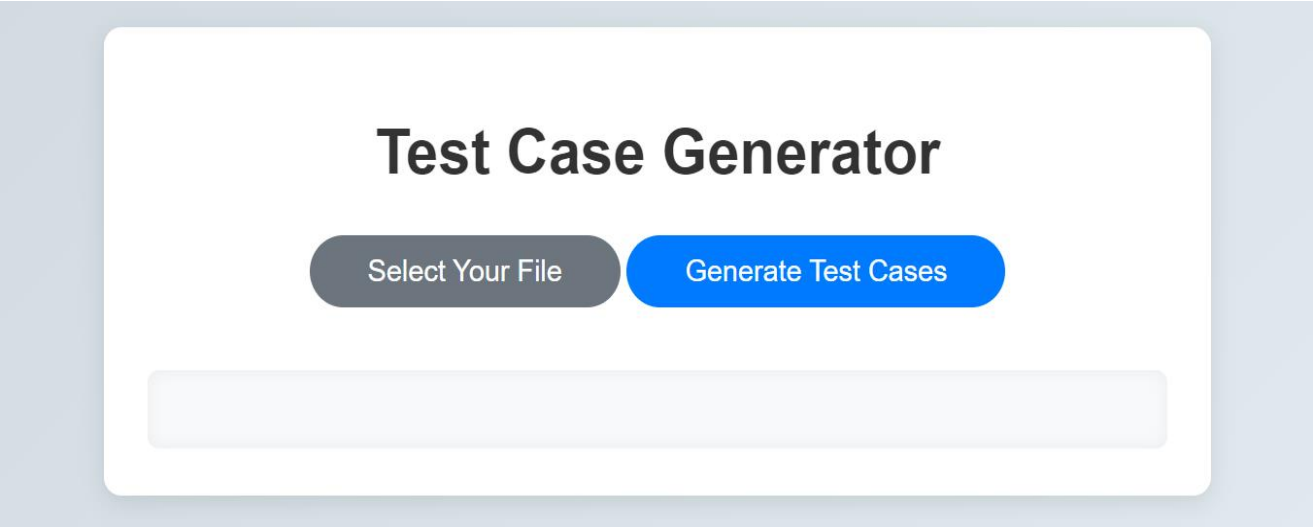
The Unit Test Case Generation application automates the process of generating unit test cases for code snippets or files written in languages like C#, Java, and Python.

The application includes a user-friendly web interface built with HTML and JavaScript, allowing users to upload code files and receive generated test cases.

OVERVIEW

FRONTEND:

Technology: HTML, JavaScript



FEATURES:

- Interface for uploading files, with support for `.cpp`, `.java`, and `.py` formats.
- Section for displaying the total number of test cases along with the generated test cases.
- Comprehensive Test Case Display
- Robust error handling for file validation and any issues that arise during response processing

BACKEND:

Technology: Python

COMPONENTS:

- File Handling: Accepts code files from users, supports reading and processing.
- Text Processing: Extracts text from the uploaded code files.
- Generative AI Integration: Uses Google Generative AI via the google.generativeai library to generate unit test cases.
- Response Handling: Parses the AI-generated response and formats it for display

APPROACH

FRONTEND:

- Create a responsive UI with HTML and JavaScript.
- Implement file upload and form submission functionalities.
- Display test cases and error messages appropriately

BACKEND:

- Set up Flask to handle file uploads and process the files.
- Integrate Google Generative AI for generating test cases.
- Handle the response from the AI and send it back to the front end.

TESTING:

- Test the file upload and text extraction for different file types.
- Validate the integration with the AI API.
- Ensure that the UI correctly displays test cases and handles errors.

OUTPUT

The user interface allows users to input code snippets either manually or via file upload and once a file is uploaded, the application automatically processes the content and generates relevant unit test cases from the bot