Project Name: Flashcard Web Application

Group Members

John-Wealth Toniloba Favour BHU/23/04/05/0020

Peace Hauwa Lumsam

BHU/23/04/05/0065

Joan Oboarekeh Obosiyo

BHU/23/04/05/0009

Faith Ajagun Abiola

BHU/23/04/10/0002

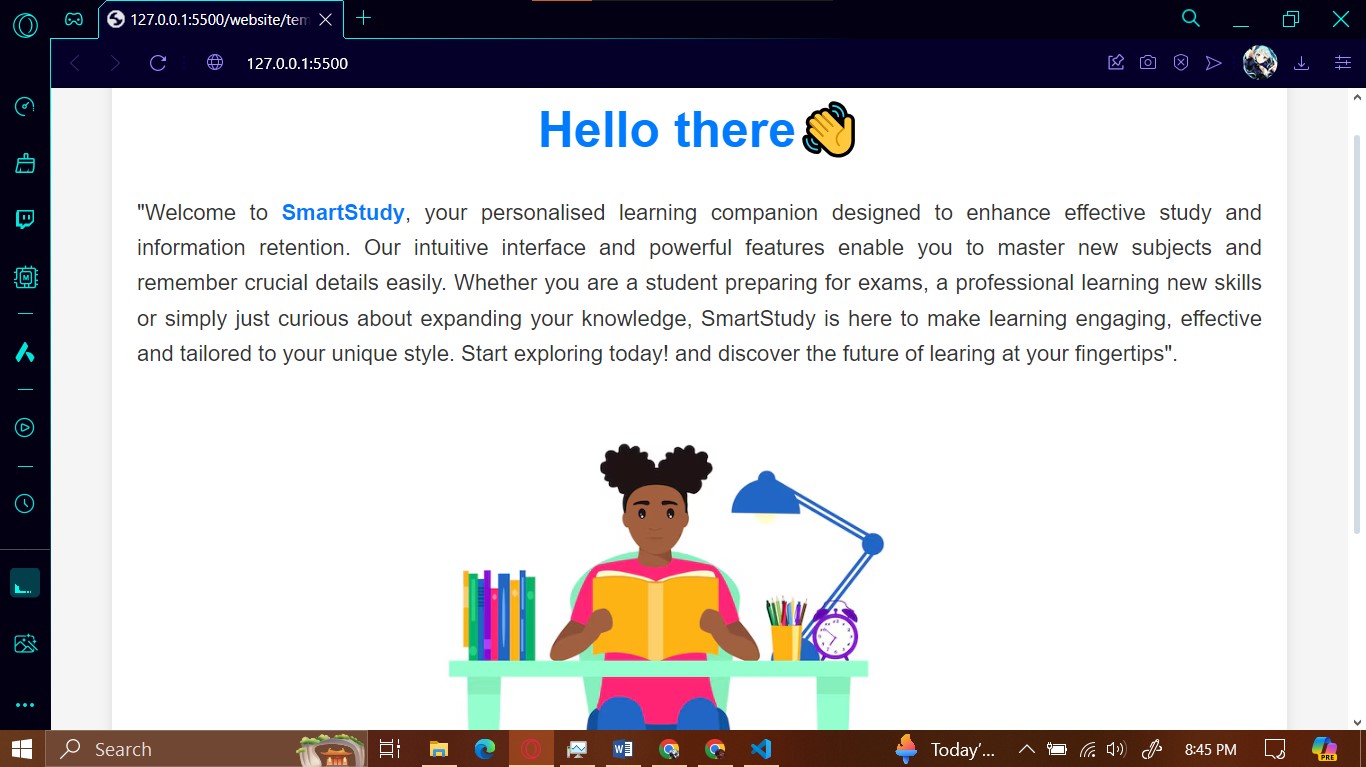
Laura Nwakra-Chimbusonma

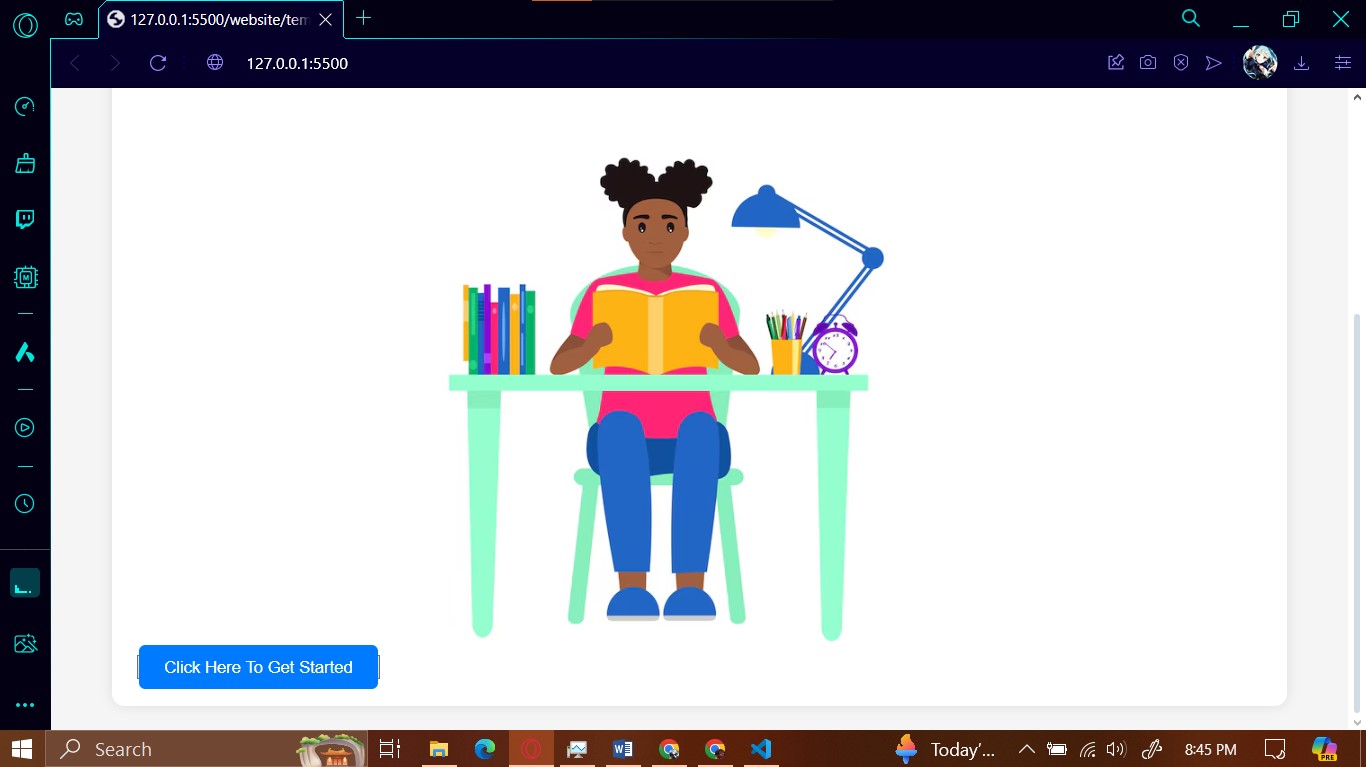
BHU/23/04/05/0084

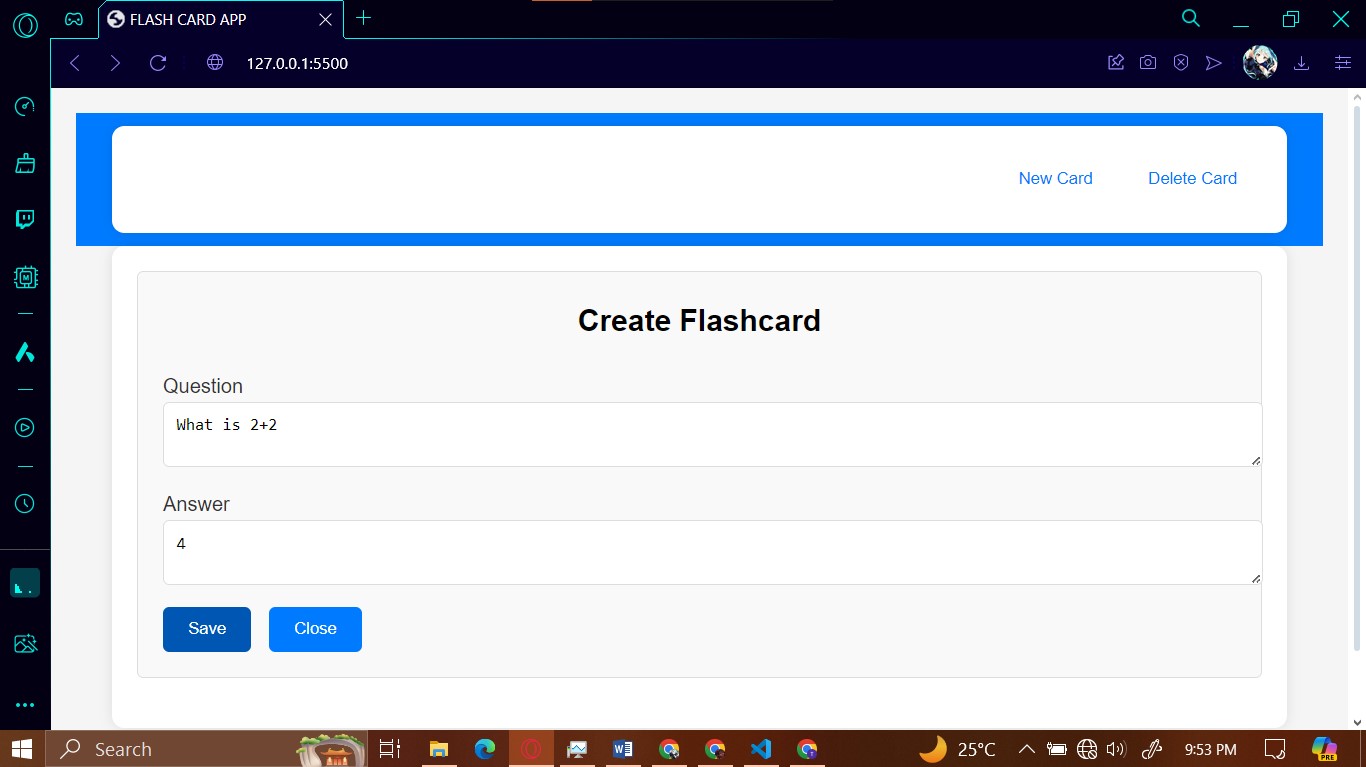
FLASHCARD WEB APPLICATION

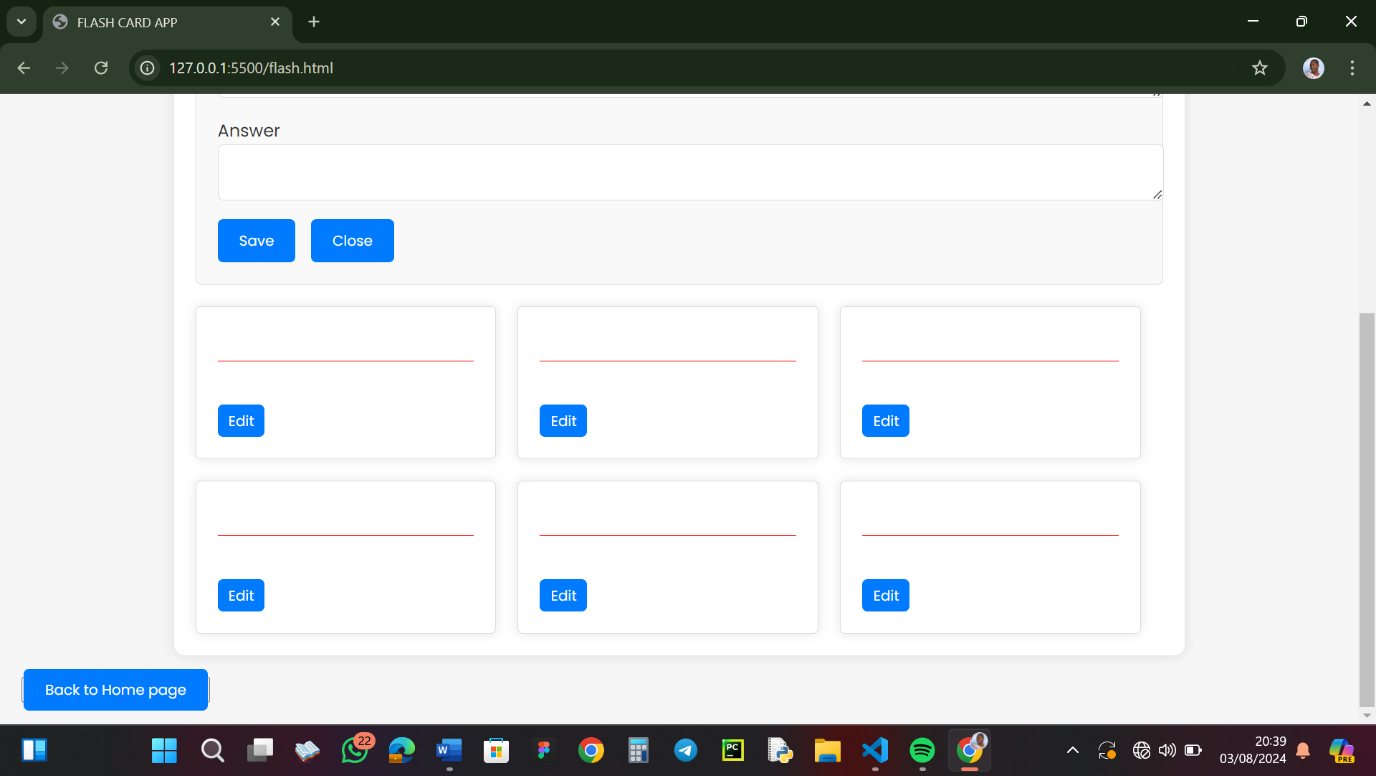
We made a flashcard web application. The Flashcard Web App is a simple and interactive web application designed to help users learn and remember information using digital flashcards. The app connects to a MySQL database to store and retrieve flashcards, making managing and expanding flashcard content easy.

It is an educational tool that enables students and teachers to create customized flashcards for a variety of subjects such as math, physics, history, and languages. Additionally, it can be utilized in corporate training environments, allowing companies to store relevant flashcards in a centralized database for efficient knowledge dissemination and skill development.









John-Wealth Toniloba Favour

BHU/23/04/05/0020

She worked on the python and flask. The app.py file is the main Python script for the web application. It serves as the backend for the flashcard app, handling requests, managing data, and rendering templates. The app.py file, powered by Flask, acts as the server-side backbone of your flashcard application, handling requests, managing data, and rendering dynamic content.

Here's a summary of the code's functionality:

1. Flask web application setup:

- Initializes a Flask server and connects to a MySQL database.

- Defines routes for index, flashcard list, add, edit, delete, and detail views.

2. CRUD operations for flashcards:

- Implements functions to create, read, update, and delete flashcards.

- Uses SQL queries to interact with the database for these operations.

3. Web interface and request handling:

- Renders HTML templates to display flashcard information.

- Processes form submissions for adding and editing flashcards.

- Manages redirects after operations and includes basic error handling.

NAME: Oboarekeh Joan Obosiyo

Matric No: BHU/23/04/05/0009

My contribution to the project focused on creating a user-centric design that supports the app's educational goals. Through careful attention to detail and a focus on usability, I helped deliver a product that not only looks good but also facilitates effective learning.

• Global Styling:

o Applied a consistent font (Poppins) and reset default browser styles for a uniform look across all elements.

o A light gray background and dark text for improved readability.

• Layout and Responsiveness:

o Designed a responsive layout using CSS Flexbox and Grid to ensure the app is accessible and functional on various devices.

o Used media queries to adjust the layout for different screen sizes, maintaining usability on mobile devices.

• Container and Typography Styles:

o Styled containers with a maximum width, padding, and subtle shadows for a professional appearance.

o Enhanced typography with bold, blue accents for titles and emphasized text for readability and engagement.

• Interactive Elements: Styled buttons and links with hover effects and transitions for a smooth and interactive user experience.

o Ensured navigation buttons were intuitive and easy to interact with through clear visual cues.

• Component Styling:

o Designed flashcard components with shadows and rounded borders to mimic real flashcards and encourage user interaction.

o Styled create and edit boxes to provide a structured and user-friendly form layout.

In conclusion, The CSS styling ensures that the app is not only aesthetically pleasing but also functional and accessible, aligning with the educational objectives of the project which is to enhance effective reading and good memorization practices.

Peace Hauwa Lumsam

BHU/23/04/05/0065

She worked on the index.html file. The index.html file serves as the entry point of the web application. It is the main HTML file that users load first when they access the flashcard app. This file sets up the foundational structure of your app and can act as the home page or dashboard, where users can access different sections of the app, like viewing existing flashcards, creating new ones, or navigating to specific study areas.

1. Structure and metadata:

- Uses HTML5 doctype and structure

- Includes meta tags for character encoding and viewport settings

- Links to an external CSS file for styling

2. Content:

- Creates a landing page for "SmartStudy", a personalized learning companion

- Contains a welcome message explaining the purpose and benefits of the platform

- Includes a title, paragraphs of descriptive text, and a call-to-action button

3. Design elements:

- Utilizes CSS classes for styling (e.g., "container", "title", "btn")

- Incorporates an image (likely a logo or illustration) related to studying

- Uses a span tag to highlight the product name "SmartStudy" within the text.

Laura Nwakra-Chimbusonma

BHU/23/04/05/0084

She worked on the flash.html file. The flash.html file is typically where the actual flashcard functionality resides. This page provides the interface for viewing, flipping, and navigating through flashcards. It interacts closely with Python and Flask to present the questions and answers dynamically, based on user interactions and data from the database.

1. Document structure and resources:

- Sets up an HTML5 document with the English language

- Links to an external CSS file and Google Fonts for styling

- Defines the page title as "FLASH CARD APP"

2. Header and navigation:

- Creates a header with a container for navigation

- Includes a title "Flashcards" and two buttons: "New Card" and "Delete Card"

3. Flashcard creation interface:

- Provides a section for creating new flashcards

- Includes a centred heading "Create Flashcard"

- Starts a form with a label and text area for entering a question

- Suggests further form elements for answers and other flashcard details (not visible in the image)

The HTML structure forms the basis of a flashcard application interface, allowing users to view, create, and delete flashcards.

Faith Ajagun Abiola

BHU/23/04/10/0002

She also worked on flash.html. She mainly worked on displaying the questions and providing an option to reveal the answer and interactive buttons, such as a "Next" button, to load the next flashcard, facilitating a smooth learning experience for users.

1. HTML Structure for Flashcard Application:

- The HTML code structures a web page with a container for displaying flashcards and a "Back to Home page" button. There are placeholders for displaying flashcards dynamically within a `<div>` with the class `flashcards`.

2. \*\*Create and Edit Flashcards:

* The code provides a user interface for creating and editing flashcards. It includes a `create-box` section for adding a new flashcard with an answer input field and buttons to save the flashcard or close the creation box.
* There's also an `edit-box` section for editing existing flashcards, with text areas for editing questions and answers and buttons to save edits or close the edit box.
* The `addFlashCards` function adds a new flashcard, while `saveEdits` updates an existing one. The code's overall structure shows that the application manages flashcards dynamically, allowing users to interact with and modify the flashcards through a web interface.

Screenshot of GitHub Contribution Page for the project

