Project Name: Online Face Recognition Attendance System

Developer's Information:

Name: Sanjida Akter Bably

Student Id: 221-15-5615

Section: 61_Q

Institution: Daffodil International University

Project Principle: Principle of Optimized Performance

This project addresses the Principle of Optimized Performance, focusing on:

- Minimizing Rebuilds
- Optimizing Rendering
- Network and Data Optimization
- Memory Management
- UI Responsiveness

Project Architecture Patterns:

To productively resolve normal issues in the programming plan, this undertaking utilizes the MVC and MVP design designs:

MVC Pattern:

The Model-View-Regulator (MVC) design is a product configuration design that isolates an application into three fundamental parts:

- **Model:** The board's and the business's reasoning are stated in the handles. It directly manages the data, reasoning, and rules of the application.
- **View:** Liable for conveying UI parts. It communicates client instructions to the regulator and displays the information for the client..
- **Controller:** Responds to UI actions and updates the view accordingly. It acts as an intermediary between the model and the view.

Example:

In a face affirmation system, the Model would manage taking care of and dealing with facial data, the View would show the client's face and cooperation status, and the Controller would regulate associations, for instance, enlisting another face or logging support.

MVP Pattern:

The Model-View-Moderator (MVP) design is a subset of the MVC design, and its main role is to coordinate the UI. It improves on testing and upkeep by isolating the application rationale from the UI:

- **Model:** Governs business behaviors and state management. It handles the data part of the application.
- **View:** Renders UI components and communicates with the presenter via an interface. It is responsible for displaying data to the user.
- **Presenter:** Goes about as a broker between the model and the view, taking care of the business rationale. It recovers information from the model and applies rationale to choose what to show in the view.

Example:

The Model would handle the information about facial acknowledgment, the View would display the results of the acknowledgment, and the Moderator would handle the information and add participation data to the view in the face acknowledgment framework.

Planning and Requirements:

The project will initially be based on the MVP architecture due to its focus on UI design and presentation layer responsibilities.

Key Features:

Face Detection: The system can detect any trained face.

Attendance Logging: Basic structure for logging attendance (future work).

User Interface: Design a user-friendly UI for ease of use.

Development Tools, Methods & Environment Explanation:

1. Development Tools:

Programming Language: Python

Framework: Django for backend development and application structure.

Face Recognition Library: OpenCV and face_recognition libraries for face detection and recognition.

Database: PostgreSQL for storing user data and face recognition data.

Front-end Technologies: HTML, CSS, and JavaScript for the user interface.

Version Control: GitHub for source code management.

2. Methods:

Development Method: Methodology for incremental and iterative development known as agile This includes normal updates, criticism cycles, and nonstop improvement of the venture.

Project Management: Jira for collaboration and project progress tracking. Sprints will be used to divide tasks, allowing for clear milestones and objectives.

3. Environment Explanation:

The development environment will consist of:

Local Development: Setting up a nearby server involving Django for improvement and testing. VSCode will be the essential IDE, outfitted with the vital augmentations for productive coding.

Version Control: Using GitHub for version control ensures that changes are tracked, and multiple team members can collaborate effectively.

Testing Environment: Separate testing environment to run unit and integration tests, ensuring the application functions correctly before deployment.

Future Features and Development of the Project:

Future enhancements will be based on user feedback and evolving requirements.

Attendance Count: Implement functionality to count and log attendance based on face recognition.

Push Notifications: Notify users of their attendance status via push notifications.

Profile Management: Users can create and manage their profiles.

Login System: Secure login with email and password.

Forgot Password: Password recovery option with email verification.

Email Verification: Verify user emails with a 6-digit PIN.

Edit Profile: Users can update their profile information and profile images.

Calendar Integration: Users can view attendance history on a calendar.

Overview of this Project and Conclusion:

The Electronic Face Affirmation Support System will start with the MVP design for beginning new development, focusing in on the UI and face recognizable proof. To manage more convoluted business rationale and state the board, it will ultimately integrate MVC engineering designs.

Using Python and Django will give a lively design to cultivating a flexible and practical application. The hidden farewell will focus on the web stage, with provisional game plans to expand handiness and back additional components as portrayed. The undertaking will be completely executed and kept up with for ideal execution and client fulfillment, despite starting impediments.

This task aims to provide an effective, efficient, and simple-to-understand framework for face acknowledgment participation, facilitating executive participation through cutting-edge innovation and plan standards. The framework's general dependability and ease of use are improved by consolidating the MVP and MVC examples to make an engineering that is adaptable, viable, and testable.