Forth Deliverable: Tools and Development tools

Isra Mariem Thabti , Hiba Abdelli BA/IT

The Facial Recognition Based Attendance System is designed to automate the process of recording attendance using facial recognition technology. This system offers a convenient and efficient alternative to traditional attendance tracking methods by eliminating the need for manual data entry.

Tools Used :

- **Frontend Tools**:
    - HTML: used for structuring the web pages and defining the layout.
    - CSS: used for styling the web elements, including colors, fonts, and layout design.
    - Bootstrap: integrated for its CSS framework to enhance the responsiveness of the web application.
- **Backend Tools**:
    - Python: The primary programming language used for backend development in this project .
    - OpenCV (cv2): Utilized for image processing tasks : face detection and recognition.
    - Flask: we have chosen it as the web framework to build and serve the web application.
    - NumPy: used for numerical computing tasks
    - Scikit-learn (sklearn): employed for implementing the **K-Nearest Neighbors** algorithm for face recognition
    - Pandas: utilized for data manipulation and analysis :   handling CSV files for attendance records.
    - Joblib: used for saving and loading machine learning models.

Development Phases:

1. **Design and Planning**:
    - The project began with defining the concept , the system's functionalities and planning the layout and features of the web application.
    - Frontend design considerations included creating a user-friendly interface for taking attendance, adding new users, and displaying the number of existing users in the database .
    - Backend functionalities were planned to include face detection, recognition, attendance recording , and model training.

2. **Backend Development**:
    - Python was used to implement the backend functionalities of the system.
    - OpenCV was integrated for face detection and recognition tasks.
    - Flask was employed to create web routes (functions) and serve HTML templates.
    - The K-Nearest Neighbors algorithm from scikit-learn was utilized for face recognition.
    - Backend functionalities included extracting faces from images, training the face recognition model, recording attendance, and managing users.
3. **Machine Learning Model Training**:
    - The machine learning model for face recognition was trained using the K-Nearest Neighbors algorithm.
    - Training data comprised images of registered users stored in a separate directory (10 screenshots of each registered user) .
    - The trained model was saved using Joblib for future use.
4. **Integration and Testing**:
    - Backend functionalities were integrated with the frontend HTML/CSS components to create a cohesive web application.
    - Testing was performed to ensure the proper functioning of face detection, recognition, attendance recording, and user management features.
    - We included some error handling mechanisms in our code in order to handle unexpected errors and edge cases gracefully.
5. **Deployment**:
    - Once development and testing were complete, the Flask application was deployed to a server for production use.

Deployment Options:

Typically, deployment options incur expenses rather than being provided for free. Consequently, we explored alternative solutions.However, it's important to note that these alternatives also entail limitations, similar to those found with **Heroku**.

Heroku:

- Popular cloud platform offering a free tier with 512 MB dyno size and support for Python applications.
- Provides a simple setup process through the Heroku CLI or web interface.
- Free tier includes intermittent sleep periods, so it might not be ideal for high-traffic applications.

Limitations of Heroku :

* Limited Dyno Hours: Free tier offers limited monthly dyno hours.

* Resource Constraints: Free dynos have limited resources, suitable for low-traffic apps.

*Add-On Costs: Some add-ons may require payment.

* Custom Domains: Custom domains may require a paid plan.

SSL Costs: SSL certificates for custom domains may incur costs.

* Support: Free tier doesn't include support from Heroku.

By leveraging a combination of frontend web technologies and backend Python frameworks, the system provides a user-friendly interface for managing attendance records and user data.