**AI-DRIVEN EYEWEAR VIRTUAL TRY-ON SYSTEM**

**ABSTRACT**

The fashion and eyewear industry has witnessed a massive transformation in recent years due to the integration of Artificial Intelligence (AI) and Virtual Reality (VR) technologies. This project focuses on developing an **AI-driven virtual try-on system for eyewear**, enabling users to visualize how different eyeglasses will look on their faces before making a purchase. The system leverages **Flask (Python)** as the backend framework, a database for managing user authentication and records, and image processing techniques for generating realistic try-on results.

The project allows users to **sign up, log in, and securely manage their profiles**, while the eyewear try-on feature uses AI-based techniques to overlay glasses on the user’s captured image. The system aims to enhance **customer experience, reduce product return rates, and improve decision-making while shopping for eyewear online**.

The project also includes an **admin panel**, similar to Django Admin, to manage users and keep track of their activities. The end result is a complete, interactive, and secure application combining **machine learning, image processing, and web development**.

**MAJOR MODULES**

1. **User Authentication Module**
   * Implements **secure login and signup** using Flask and SQLAlchemy.
   * Stores **hashed passwords** to ensure high security.
   * Prevents unauthorized access by redirecting unregistered users to the login page.
   * Session management ensures only authenticated users can access the try-on feature.
2. **Virtual Try-On Module**
   * Users can **upload or capture their photo**.
   * The system processes the image and **overlays eyewear frames** using AI and image processing techniques.
   * Generates **three different try-on results** with different eyewear models.
3. **Admin Panel Module**
   * Built using **Flask-Admin**, which functions similarly to Django Admin.
   * Allows the administrator to view, update, or delete user records.
   * Useful for managing user activities and system monitoring.
4. **Image Upload and Processing Module**
   * Allows users to **upload their face images** in formats like .jpg or .png.
   * Implements a function to **clear previously generated results** to avoid clutter.
   * Generates multiple try-on results dynamically.
5. **Result Display Module**
   * Displays the processed images in a visually appealing format.
   * Provides users with three eyewear try-on options for better comparison.

**MINOR MODULES**

1. **Flash Messaging System**
   * Provides users with instant feedback, such as:
     + “Login Successful!”
     + “Invalid Username or Password”
     + “Signup Successful!”
2. **Session Handling**
   * Automatically logs out users after session expiry or manual logout.
3. **Error Handling**
   * Handles missing file uploads, incorrect credentials, and server-side issues gracefully.
4. **UI Enhancement Module**
   * Neon-themed user interface designed with **HTML, CSS, and animations**.
   * Provides an engaging and modern look.
5. **Database Initialization Module**
   * Creates tables automatically when the project is run for the first time.

**TOOLS USED**

1. **Backend Framework:**
   * **Flask** (lightweight, fast, and flexible).
2. **Database:**
   * **SQLite** (lightweight relational database, integrated with SQLAlchemy ORM).
3. **Admin Panel:**
   * **Flask-Admin** for easy database record visualization.
4. **Frontend Tools:**
   * **HTML5** for structure.
   * **CSS3** for styling and animations.
   * **JavaScript (optional)** for interactive behavior.
5. **Image Processing:**
   * **OpenCV / Custom ML Model (in model.py)** for eyewear overlay.
6. **Security Tools:**
   * **Werkzeug’s Password Hashing** for secure authentication.
7. **IDE and Development Tools:**
   * **VS Code / PyCharm** for development.
   * **Postman** (optional) for API testing.

**DATABASE**

* **Database Name:** eyewear.db
* **Table Name:** User
* **Fields:**
  + **id** (Primary Key, Integer)
  + **username** (Unique, String)
  + **email** (Unique, String)
  + **password** (Hashed String)

Example Record:

| **ID** | **Username** | **Email** | **Password (Hashed)** |
| --- | --- | --- | --- |
| 1 | john123 | john@email.com | pbkdf2:sha256$... |

The database is automatically created when the application is run for the first time. All users are displayed in the admin panel.

**ER DIAGRAM**

pgsql

CopyEdit

+-------------------+

| User |

+-------------------+

| id (PK) |

| username |

| email |

| password (hashed) |

+-------------------+

|

| 1 : N

|

+---------------------+

| Try-On Results |

+---------------------+

| result\_id (PK) |

| user\_id (FK) |

| image\_path |

| created\_at |

+---------------------+

The relationship indicates **one user can have multiple try-on results**.

**DESCRIPTION**

The **AI-Driven Eyewear Try-On System** is designed to solve the problem of online eyewear shopping, where customers are often hesitant due to uncertainty about how glasses will look on them.

**Workflow:**

1. A user signs up and logs in.
2. After successful login, the user is redirected to the **index page**.
3. From the index, the user navigates to the **capture page** to upload or capture a photo.
4. The photo is processed by the generate\_tryon\_images() function, which overlays three different eyewear designs on the user’s face.
5. The processed images are displayed on the **results page**.

**Key Features:**

* Smooth user experience with interactive UI.
* Secure authentication.
* Admin panel for monitoring users.
* Real-time try-on results generated dynamically.

This project demonstrates how **AI and web technologies can blend seamlessly to enhance e-commerce experiences**.

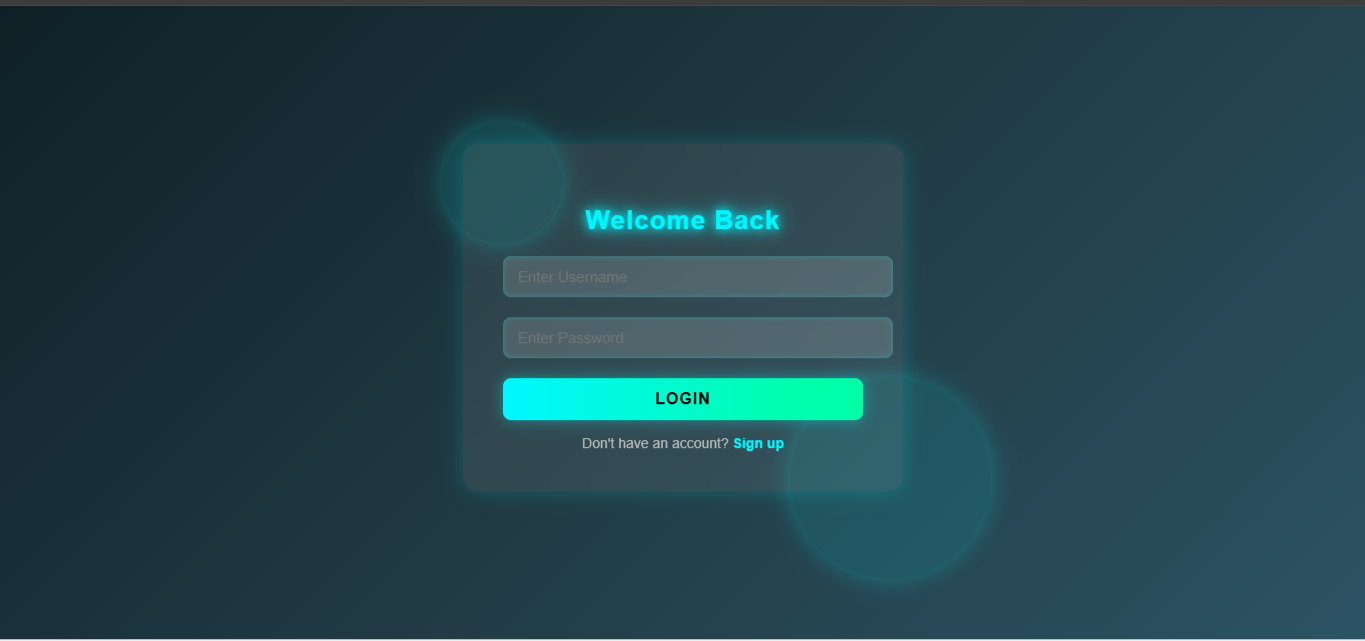
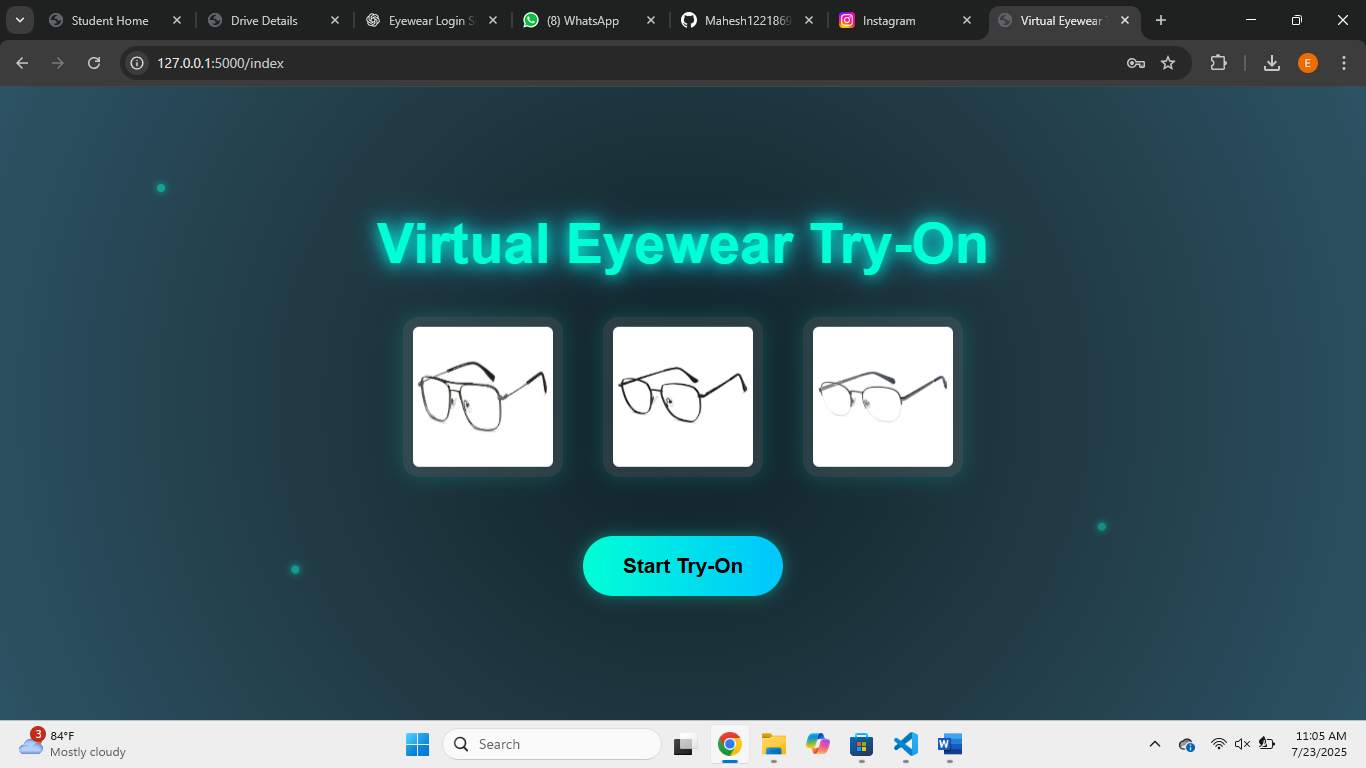
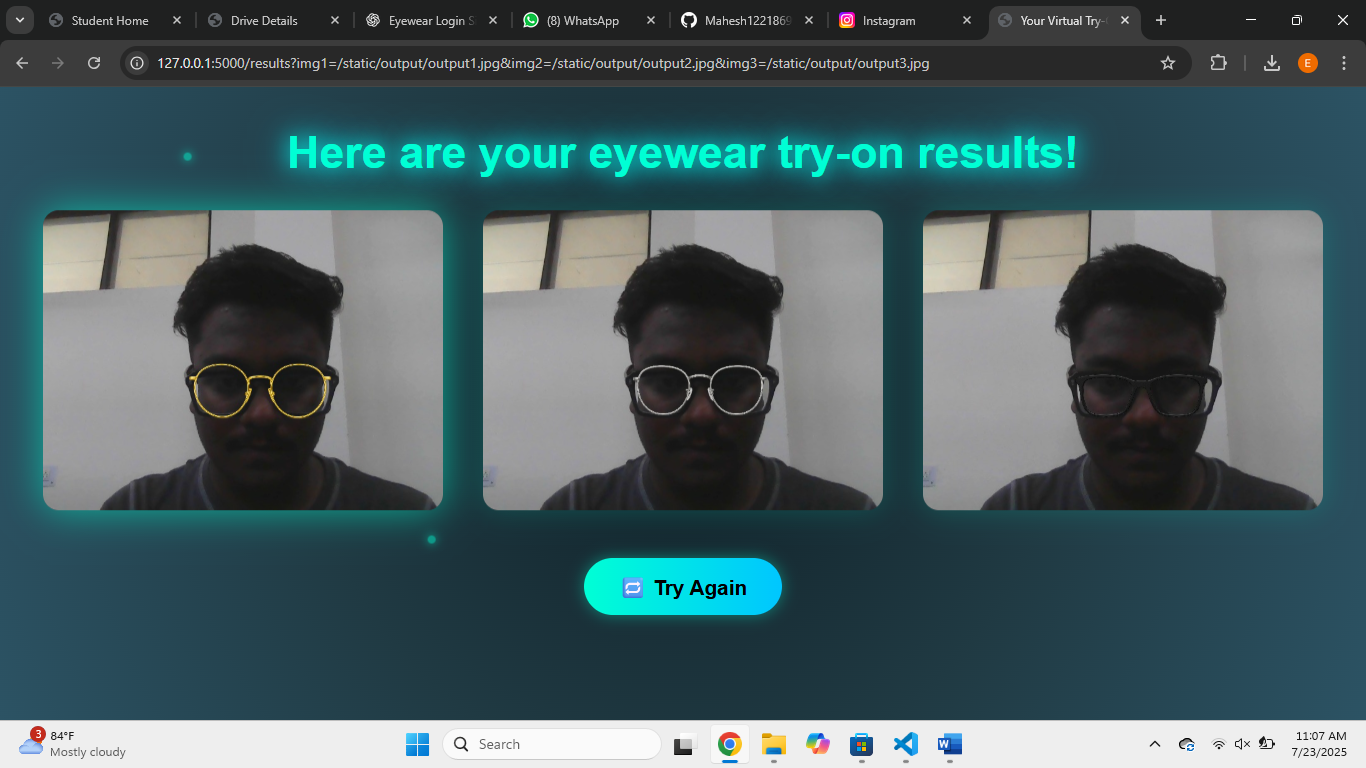
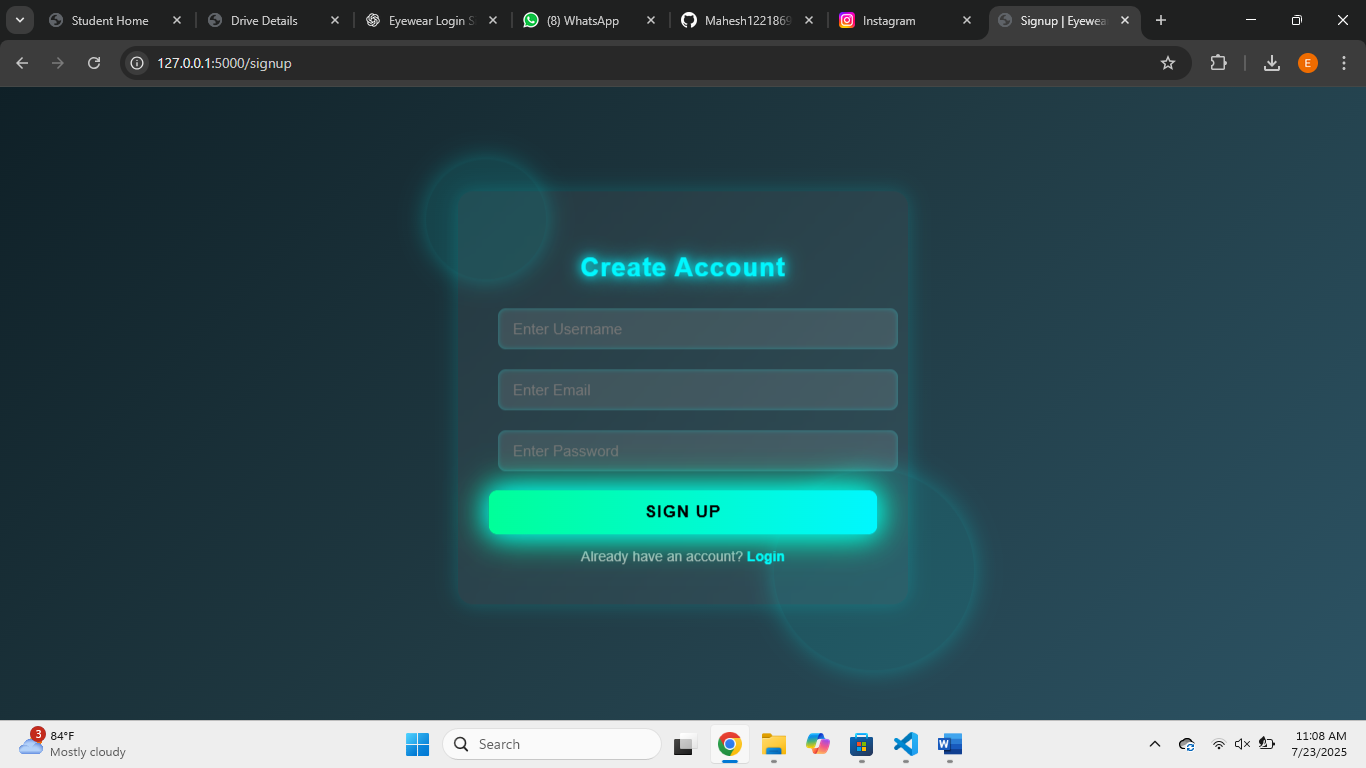
**SCREENSHOTS *(Add real screenshots when preparing the final document)***

1. **Login Page** – Neon-styled login screen with animated effects.
2. **Signup Page** – Secure user registration form.
3. **Admin Panel** – Displays all registered users.
4. **Index Page** – Dashboard for navigation.
5. **Capture Page** – Upload/capture image.
6. **Results Page** – Shows three eyewear try-on results.

**FEATURE ENHANCEMENT (FUTURE WORK)**

1. **Advanced AI Models:**
   * Use **3D face detection** for more realistic try-on effects.
2. **Eyewear Catalog Integration:**
   * Allow users to select from a **large catalog of eyewear frames**.
3. **E-Commerce Integration:**
   * Direct purchase links to online stores.
4. **Face Shape Detection:**
   * Recommend frames based on face shape.
5. **Real-Time Try-On:**
   * Live camera-based try-on using **augmented reality (AR)**.
6. **Social Media Sharing:**
   * Let users share try-on results on Instagram, Facebook, etc.

**SCREENSHOTS:**

****

**CONCLUSION**

The AI-Driven Eyewear Try-On System successfully combines Flask-based web development, secure authentication, and AI-driven image processing to provide users with a realistic virtual try-on experience. The project reduces product return rates, enhances customer satisfaction, and can be scaled further with AR integration and e-commerce platforms.

This project demonstrates the power of AI in retail applications and paves the way for developing smart, interactive, and personalized online shopping solutions.