

Virtual Dress System Project

Research Document

Project Overview

- The virtual dress system allows users to try on clothes virtually using their digital avatars.
- This project combines computer vision and web development technologies to provide an interactive user experience.
- The project will use Python as the backend language, with Flask for user registration and login, and OpenCV for virtual try-on features.

Objectives

- Develop a user-friendly web application for virtual dress trials.
- Implement secure user registration and login functionality.
- Use computer vision to map virtual clothes onto user avatars.
- Ensure the system is scalable and can handle multiple users.

Technologies and Tools

1. Backend

- Python: Primary programming language.
- Flask: For web framework, handling user authentication, and routing.

2. Frontend

- HTML/CSS/JavaScript: For the basic structure, styling, and interactivity of the web pages.
- React.js: For a more dynamic and responsive user interface.

3. Computer Vision and Image Processing

- OpenCV: For image processing and manipulation.
- MediaPipe : For body pose detection and alignment.
- Dlib: For facial landmark detection (if necessary).

4. Database

- MySQL: To store user information and clothing data.

Development Roadmap

1. Phase 1: Planning and Setup

Requirement Analysis:

- Identify core features: user registration, login, uploading user photos, selecting virtual dresses, etc.
- Determine the dataset for clothes (images, 3D models).
- Define system architecture and technology stack.

Environment Setup:

- Set up development environment with necessary tools and libraries.
- Initialize version control (GitHub/GitLab).

2. Phase 2: Backend Development

User Authentication:

- Implement user registration and login using Flask.
- Secure the application with proper authentication (JWT tokens or Flask-Login).

Database Integration:

- Design and create the database schema (tables and rows).
- Implement database models for user data and clothing items.

3. Phase 3: Frontend Development

Basic UI/UX Design:

- Develop basic frontend using HTML, CSS, and JavaScript.
- Create templates for registration, login, and main application pages.

Responsive Design:

- Ensure the frontend is responsive and works well on various devices.

4. Phase 4: Computer Vision Integration

User Image Processing:

- Implement functionality to upload user images.
- Use OpenCV and MediaPipe to detect body pose and landmarks.

Clothing Overlay:

- Develop algorithms to overlay virtual clothes on user images.
- Ensure proper scaling and positioning of clothes based on detected landmarks.

Real-time Try-on Feature:

- Implement a feature for real-time try-on using webcam feeds.

5. Phase 5: Testing and Optimization

Testing:

- Conduct unit tests and integration tests.
- Perform user acceptance testing to gather feedback.

Optimization:

- Optimize image processing algorithms for better performance.

- Ensure the application is scalable and can handle multiple users.

6. Phase 6: Deployment and Maintenance

Deployment:

- Deploy the application on a cloud platform (AWS, Heroku, etc.).
- Set up a domain and SSL certificates.

Maintenance:

- Monitor the application for any issues.
- Regularly update the system with new features and security patches.
- Research and Development Considerations

Security:

- Ensure all user data is securely stored and transmitted.
- Implement proper authentication and authorization mechanisms.

References

OpenCV Documentation: <https://opencv.org/documentation/>

Flask Documentation: <https://flask.palletsprojects.com/>

MediaPipe Documentation: <https://google.github.io/mediapipe/>

SQLAlchemy (for database interaction): <https://www.sqlalchemy.org/>

This document provides a comprehensive overview of the project, the technologies involved, and a detailed roadmap for development. Each phase builds on the previous one, ensuring a structured and methodical approach to the project.