Software Requirements Specification (SRS)

Virtual Glasses Try-On System

Prepared For: Continuous Assessment 3

Course: INT219 – Front End Developer

Course: INT220 – Server Side Scripting

Semester: Spring 2025

Prepared By

Name	Student ID
Avinash Singh	12408446
Vipasana Kumari	12408447
Mohammed Riyaz	12405607
Umashankar Kumar	12415406

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of the Virtual Glasses Try-On system. It serves as a guide for developers, testers, and reviewers evaluating this project.

1.2 Scope

The system is a web-based augmented reality (AR) application that enables users to virtually try on different eyeglass frames using uploaded photos. Technologies such as HTML, CSS, JavaScript, and model.js (MediaPipe Face Mesh) are used for real-time face landmark detection and overlay rendering.

1.3 Definitions, Acronyms, and Abbreviations

AR: Augmented Reality
UI: User Interface
JS: JavaScript
SRS: Software Requirements Specification

1.4 References

MediaPipe Face Mesh Documentation
W3C HTML5 & CSS3 Specifications

1.5 Overview

This document outlines functional and non-functional requirements, constraints, and architectural models needed to implement the Virtual Glasses Try-On system. It also includes diagrams, assumptions, and software dependencies.

2. General Description

2.1 Product Perspective

The application is a standalone, client-side web system requiring users to upload images. It uses facial landmark detection to overlay virtual glasses onto the user's face.

2.2 Product Functions

- Upload photo for processing
- Detect facial landmarks using model.js
- Overlay selected glasses on the detected face
- Allow users to switch between different frames
- Option to download the final image

2.3 User Characteristics

- Basic understanding of web browsers
- Open to users of all age groups

2.4 General Constraints

- Requires stable internet connection
- Compatible with modern browsers (Chrome, Firefox, Edge)

2.5 Assumptions and Dependencies

- Users will grant memory access for image upload
- Internet access available to load external libraries

3. Requirements

3.1 Functional Requirements

- The system must detect facial landmarks in uploaded photos.
- It must accurately position and scale virtual glasses based on eye position.
- Users must be able to switch between multiple glasses designs.
- The overlay should adapt to facial structure in real time.

3.2 Non-Functional Requirements

- Low latency for smooth user interaction
- Cross-browser compatibility
- Responsive design for desktop and tablets
- Intuitive and user-friendly interface

3.3 Hardware Specifications

- Internet-connected computer or smartphone
- Minimum 4GB RAM for optimal performance

3.4 Software Requirements

- Any modern web browser (Chrome, Firefox, Edge)
- Code Editor (e.g., VS Code)
- MediaPipe Face Mesh via model.js
- No backend server; all processing occurs client-side

4. Analysis Models

4.1 Data Flow Diagram (Level 0)

User → [Open Webpage] → [Upload Photo] → [Face Landmark Detection] → [Glasses Overlay] → Display Result

4.2 Methodology

4.2.1 Face Detection

Utilizes MediaPipe Face Mesh to identify 468 facial landmarks, especially eye regions.

4.2.2 Glasses Placement

JavaScript calculates coordinates and scales images based on the distance between eyes.

4.2.3 Overlay Rendering

Virtual glasses are rendered onto a canvas overlaid on the video feed or uploaded image.

4.2.4 User Interaction

Interactive buttons allow users to choose from multiple glasses options.

5. Github Repository

🔗 https://github.com/Abhiefab/virtual-try-on

6. Social Media Handles

🔗 https://youtube.com/@avinashsingh-s7x?si=UMf4SiKuGgEZXkNC

🔗 https://youtube.com/@jvipasana?si=qHDaRSiWtm8Bfh6p

🔗 https://www.youtube.com/@MohammedRiyaz-225

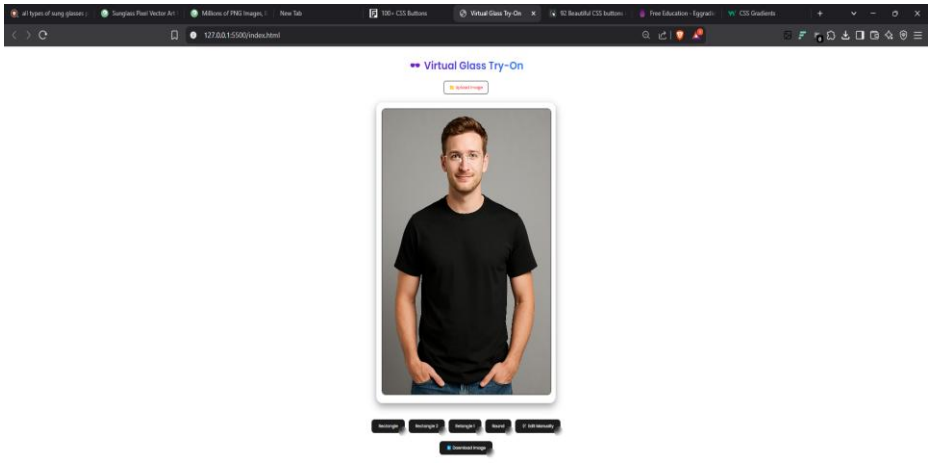
🔗 https://youtube.com/@umashankarkumar8325?si=S0GK4Tq-Xyi70t67

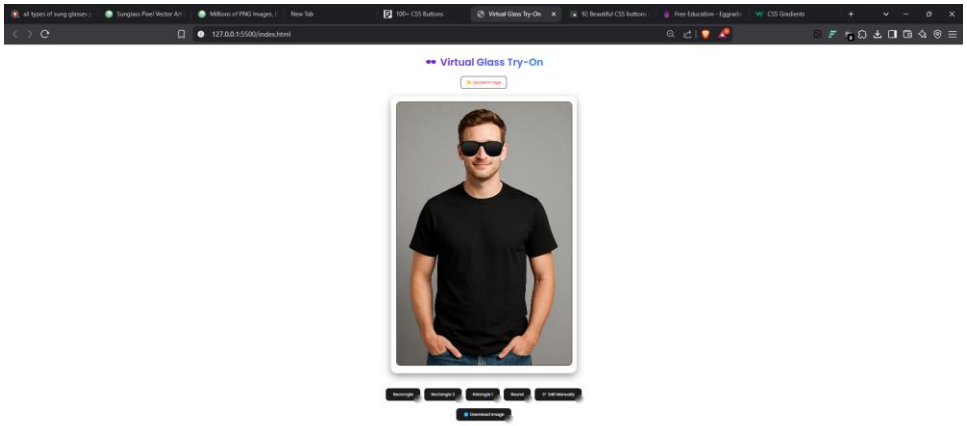
7. Appendices

A.1 Screenshots

• User interface preview

• Sample outputs with various glasses styles




Virtual Glass Try-On