Release Plan – CSE 115a – Software Engineering

Product Name: Dress to Impress

Team Name: Wired

Release Name: Star

Release Date: December 3, 2025

Revision Number: 0

Revision Date: October 12, 2025

High Level Goals

App Infrastructure and UX:

- Secure web app that protects private information
- Visually pleasing UI/UX design that entices users

Database Creation:

- Upload and digitally manage user's own wardrobe
    - Label clothes as shirts, pants, etc.
    - Label clothes based on occasion (casual, business, going out)
    - Label clothes as favorite by hearting them
- Mark outfits as worn in the database and have the ability to save that combination of clothes for future use in the database
    - Indicate worn frequency for a clothing item or outfit from information in database

AI Image Generation:

- Submit and save body profile photo for visualization
- Virtual try-on by generating realistic images showing how selected outfit looks on your body
    - Users can curate new outfits based on their existing clothing and "try them on"

Gen AI Clothing Suggestions (Stretch Goals):

- Weather tab/icon/ui change on app
- Gen AI suggestions on what clothing to wear based on weather and occasion
- Recommend something in your closet or get rid of it, if not worn very

frequently

Additional Goals:

- Outfit wrapped (spotify dupe) on Profile page

User stories defining the scope of the release:

(Initial Ideas)

- **As a new user, i want to be able to create an account**
- **As a returning user, i want to be able to log into my existing account**
- **As a user, I want to customize my profile with AI suggestions and insights about the user's wardrobe/wearing habits**
- **As a user, I want to be able to view my wardrobe pieces in a "wardrobe" page for easy and remote access to my clothing items**
- **As a user, I want to create outfits based on my wardrobe pieces and view them later in a separate tab called Outfits**

As a forgetful person, I want to make the best use of the wardrobe I have curated by having a database of all the clothes I have.

- Allow user to upload images of their clothes
  - Have the user take a photo of a clothing item with a solid color background, so the program can remove it, leaving the item on its own
- Create a database to store images of all of the clothes a user owns
- allow user to tag their clothing (shirt, pants,..)
- Allow user to tag the clothing as

I have a hard time visualizing what outfits would look like on me without putting it on, so I want an app that will be able to render something to help with that.

- Submit and save body profile photo for visualization
- Virtual try-on by generating realistic images showing how selected outfit looks on your body

As someone who wants to elevate their wardrobe, I need help assembling different outfits and suggestions on what I need for my closet so that I can develop my personal style. (Stretch goal)

- Suggest outfits to buy and what clothing items can be paired with what accessories
    - Our app needs to use AI to understand how layering of clothing works, how colors work, understand different parts of an outfit

Tasks for each sprint (Initial Plan):

Sprint 1 – Project Setup & Core Infrastructure:
- Set up github, frontend, connect to backend
- auth
- **User Story 1.1**: As a new user, I want to create an account and securely log in so that I can save and access my wardrobe and outfits.
    - Task 1: design login and signup pages
    - Task 2: implement authentication
    - Task 3: create user database schema
- **User Story 1.2**: As a user, I want a visually appealing home page with clear navigation so that I can easily access my wardrobe, outfits, and app's other features.
    - Task 1: create a main dashboard UI skeleton
    - Task 2: implement navigation bar and route management
    - Task 3: add basic responsive design

Sprint 2 – Wardrobe Upload & Tagging System

- As a user, I want to be able to upload images of my clothing pieces, and save them so I can view them later in my closet. **(database of clothes)**
- As a user, I want to be able to 'tag' each clothing item with a label describing it (ex. dressy, casual, …), to later reference when I am looking for specific outfits. **(organization of clothes)**

Sprint 3 – AI Image Generation
- **User Story 3.1** As a user, I want to be able to visualize my own clothing on myself so that I can try on clothes without going through the effort of putting them on
    - Generate outfit images with clothing combinations selected by user
    - Task 1: Set up OpenAI API key
    - Task 2: (Compositing) Prompt engineer so that background of the clothing item is removed
    - Task 3: Save the cropped image in a new table in the database with the name as the uncropped image but with a slightly different keyword in the name.
- **User Story 3.2** As a user, I don't want to type out my clothing item's information into the app so that I am not wasting my time

- [Continuation of Sprint 2's tagging system] Scan tag on outfit and fill in tagging system information from previous system
    - If unknown, prompt the user to fill in the information
    - If they don't fill it out, remind them in 2 day increments?
  - Task 1: camera/scan module: Scan tag to get information for materials, which is saved in the database in a new table as scanned
- **User Story 3.3** As a user, I want to save the outfits that have been generated so that I can look back on them later for outfit inspiration.
  - Save outfits generated from my closet from ai generated images and save those combinations
- **[SPIKE] User Story 3.4** As a user, I want to be given ideas on what to wear based on things in my closet
  - Generate outfits from existing pants, shirts, accessories based on information from tagging system (only focus on dress code)
  - Task 1: take in user's measurements and recommend silhouettes so we don't have to use an image of the users body photo

Spike: mnist dataset with tensorflow, dataset with images of clothing, used to train model so that model can sort clothes instead of asking user to identify clothing.

Sprint 4 – AI Suggestions & Stretch Features
- Weather API
- AI Recommendations
- **User Story 4.1** ("As a user, I want outfit recommendations based on weather and occasion so that I can plan what to wear.")
  - Task 1: Integrate Weather API and parse weather data (3 h)
  - Task 2: Implement logic to map temperature/conditions to outfit tags (3 h)
  - Task 3: Build AI prompt generator (e.g., rules + GPT API placeholder) (3 h)
  - Task 4: Design and implement "Suggested Outfits" UI cards (3 h)
- Total: 12 hours
- **User Story 4.2** ("As a user, I want to see which clothes I rarely wear so that I can declutter.")
  - Task 1: Query database for low-frequency items (3 h)
  - Task 2: Display results on dashboard (3 h)
- Total: 6 hours
- **User Story 4.3** ("As a user, I want a reliable and consistent app experience so that I can trust it daily."
  - Task 1: Perform end-to-end testing (3 h)
  - Task 2: Fix bugs and other errors (3 h)
  - Task 3: Finish final documentation before deployment (3 h)

Initial Presentation:

Sanity Check:

We are confident in our abilities to adapt to new technologies and achieve our MVP. Our work distribution is reasonable, with the majority of the work being in sprints 2 and 3, which happen before holidays. Sprint 4 will be the time when we work on our stretch goals and add finishing touches to the product. We have also spent a lot of time researching technologies and setting up the infrastructure for this project.

Product Backlog:

A listing of all high-level goals and user stories that were discussed in
the release planning meeting, but which did not make it into the release at this point. User story priorities may change in the course of the project and therefore the PO may decide to downgrade some user stories currently in the release plan and promote some user stories currently in the backlog. The release plan and product backlog should be revisited and updated after each sprint.
The product backlog remaining at the end of the last sprint can serve as the starting point for a subsequent release.

- ("As a user, I want outfit recommendations based on weather and occasion so that I can plan what to wear.")
- ("As a user, I want to see which clothes I rarely wear so that I can declutter.")
- ("As a user, I want a reliable and consistent app experience so that I can trust it daily.")