Sprint 2 Plan
Product: Dress To Impress
Team: Wired
Completion Date: 10/24/25
Revision Number: 1
Revision Date:

Goal:
Frontend:
Profile Page

- User info
- Wardrobe stats
- Wearing history
- Forgotten Finds

Wardrobe Page

- Create files to setup overall framework
- Create profile page to start connecting API routes to print out information
- Nav Bar component
- Item component
- Add item page
- Item insight page with larger image, tags
- Filter search on wardrobe page

Outfits page

- No outfit created yet
    - Start your collection button
- Creating an outfit
    - Swipe through shirts and pants
    - Add item button (with pop up of certain things grayed out if already selected)
- Outfit database
    - Scroll through collections of outfits
    - Shirt + Pant minimum, 3rd item = outerlayer (BIG), 3rd item = accessory (small), 3+ = go to more page
    - More page is just a page where you can scroll through the items selected like in the wardrobe page
    - Button in the top right corner to edit outfit (remove pieces)

Backend:

- load databases with sample data
- create api routes for auth and clothing
- integrate closet and outfit database with frontend

Sarone:
Sofia:

Shivani:
Jenny:
Sona:

User Stories

- **User Story 2.1 As a new user, I want to be able to create an account (3 points)**
  - Frontend:
    - Task 1: Create sign up
  - Backend:
    - Task 1: Research how to run the frontend without signing in repeatedly. Do research on the RSL policies and what we need to change the service role key to when we are done.
    - Task 2: create api routes for user creation to supabase auth
    - Task 3: add user data and test api calls
    - Task 4: connect to the frontend
- **User Story 2.2 As a returning user, I want to be able to log into my existing account (3 points)**
  - Frontend:
    - Task 1: Create log in
  - Backend:
    - Task 1: Create api routes to check for user information in supabase auth
    - Task 2: add user data and test api calls
    - Task 3: Connect to the frontend

- **User Story 2.3 As a user, I want to be able to upload and save my clothing pieces and their images. (5 points)**
- Frontend:
  - ~~Task 1: "Add Item" form to upload an image that can be saved and make sure that items can be tagged based on the labels in supabase~~
  - Task 2: implement file upload handler
  - ~~Task 3: Make sure all labels that can be selected by user match up with what is in supabase~~
- Backend:
  - ~~Task 1: upload test data with images in the supabase tables for clothing_items~~
  - ~~Task 2: implement RLS policies~~
  - ~~Task 3: integrate upload form -> Supabase upload + DB insert~~
  - Task 4: create api routes to retrieve items and images
  - ~~Task 5: Update navigation so "Add Clothing" is accessible from Closet home~~

- **User Story 2.4 As a user, I want to view my clothes in my closet. (8 points)**
  - **Frontend**
    - Task 1: create reusable image card component
    - Task 2: Build category grid page and navigation to closet/[category]
    - Task 3: Create dynamic category page to fetch items filtered by category
  - Backend:
    - Task 4: connect category pages -> Supabase query + signed URL generation
    - Task 6: integrate route to get images from bucket
    - Task 7: Verify that all clothing items are displayed on wardrobe page

- **As a user, I want to be able to edit individual saved clothing items' labels (ex. dressy, casual, …). (organization of clothes) (3 points)**
  - Frontend:
    - Task 1: add edit functionality
    - Task 2: add an edit form
  - Backend:
    - ~~Task 1: Create api route that allows user to change an entry of database~~
    - Task 2: integrate edit form with supabase query

- **As a user, I want to view clothes side by side in the outfits page to see how they look together (8 points)**
  - Frontend:
    - ~~Task 1: make sure existing outfit in the backend that I added in sprint 1 shows up in the outfits tab~~
    - Task 3: create frontend component for a bar that shows up at the bottom when a clothing item is selected in the wardrobe page
    - Task 1: create system so that once a clothing item is selected in the wardrobe page, a new button called create outfit pops up
  - Backend:
    - Task 1: reuse api routes to retrieve items and images
    - task 2: integrate with frontend
- **As a user, I want to create outfits based on my wardrobe pieces (3 points)**
  - Frontend:
    - Task 1: create save button on the outfits page
    - Task 2: create(reuse) individual item display modal
  - Backend:
    - ~~Task 1: create api routes to get outfit data to outfit join tables~~
    - Task 2: create api route to post outfit data once that data is received in the frontend
    - task 2: integrate with frontend