Backend Planning Meeting Summary

# Next In Person Meeting:

Next week 2/5/2025 – 11:00am to 1pm

# Immediate Priorities

|  |  |  |
| --- | --- | --- |
| Task | Responsible | Due |
| Finish html templates (analysis.html, social.html, intro.html, personal.html) | Duc and Ralisha | Tonight |
| Integrate Jinja into all templates | Prem | After templates are done |
| Start backend setup | Everyone | Tomorrow |
| Complete user authentication with Flask session | (anyone) | Friday |
| Finalise database schema (ERD + models.py) | Duc | Sunday night |

# Agreed Backend Technologies

- Flask sessions for login/signup

- SQLite3 as the database

- rembg or transparent-background for image processing

- WTForms for input validation

- models.py for DB models

- utils.py for reusable backend functions

- database.db as the main DB file

# Backend Functionalities to Implement

## Upload Clothing

- Make uploaded image transparent

- Save file to uploads/user\_<id>/...

- Store item info (type, name, color, tags) in DB

## Wardrobe View

- Display only current user’s clothes

- Include item name/type visibly

- Allow filtering by season/type/etc.

## Outfit Generator

- Let user pick: Random or Filter by season + occasion

- Algorithm randomly selects items from DB

- Displays in a preview panel

## Save Outfit

- Save selected clothes into an outfit

- Link outfit ↔ clothing via foreign keys

- Add outfit name + public/private toggle

## Outfit Gallery

- Show saved outfits

- Filterable by occasion/season

## Social Page

- Shows all public outfits + user’s name

- Visible only if marked public

## Analysis

- Show: Total clothing items, Total outfits, Most-used item

- Use charts (e.g., bar, pie) to visualise insights

## Extra Needs

- Error validation for image uploads

- Route protection via session check

- Proper model relationships in DB

- CRUD for deleting items/outfits

# Flask File Structure Plan

app/  
├── static/uploads/user\_<id>/  
├── templates/  
│ ├── base.html  
│ ├── wardrobe.html  
│ ├── outfit.html  
│ ├── analysis.html  
│ ├── social.html  
├── fashanise.py  
├── \_\_init\_\_.py  
├── routes.py  
├── models.py <-- DB models  
├── utils.py <-- helper funcs (e.g., rembg)  
├── database.db <-- SQLite

# Timeline (Short-Term)

|  |  |
| --- | --- |
| Task | Deadline |
| User auth with Flask sessions | Friday |
| DB structure (ERD + models.py) | Sunday |
| Upload logic, file saving, rembg | Early next week |
| Outfit generator logic + preview | After DB done |
| Analysis & charts | Post-DB + WTForms |