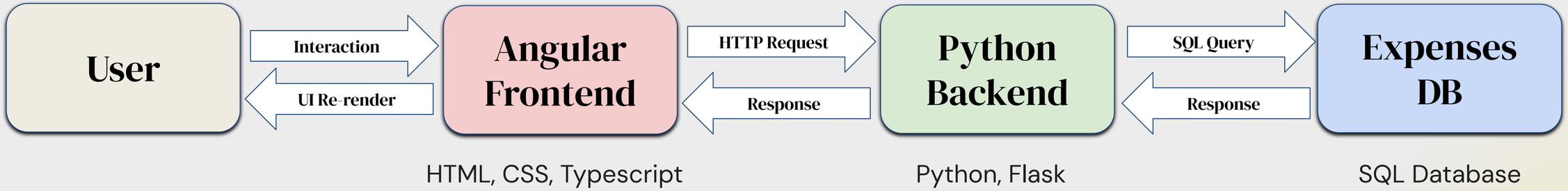
Expense Tracker

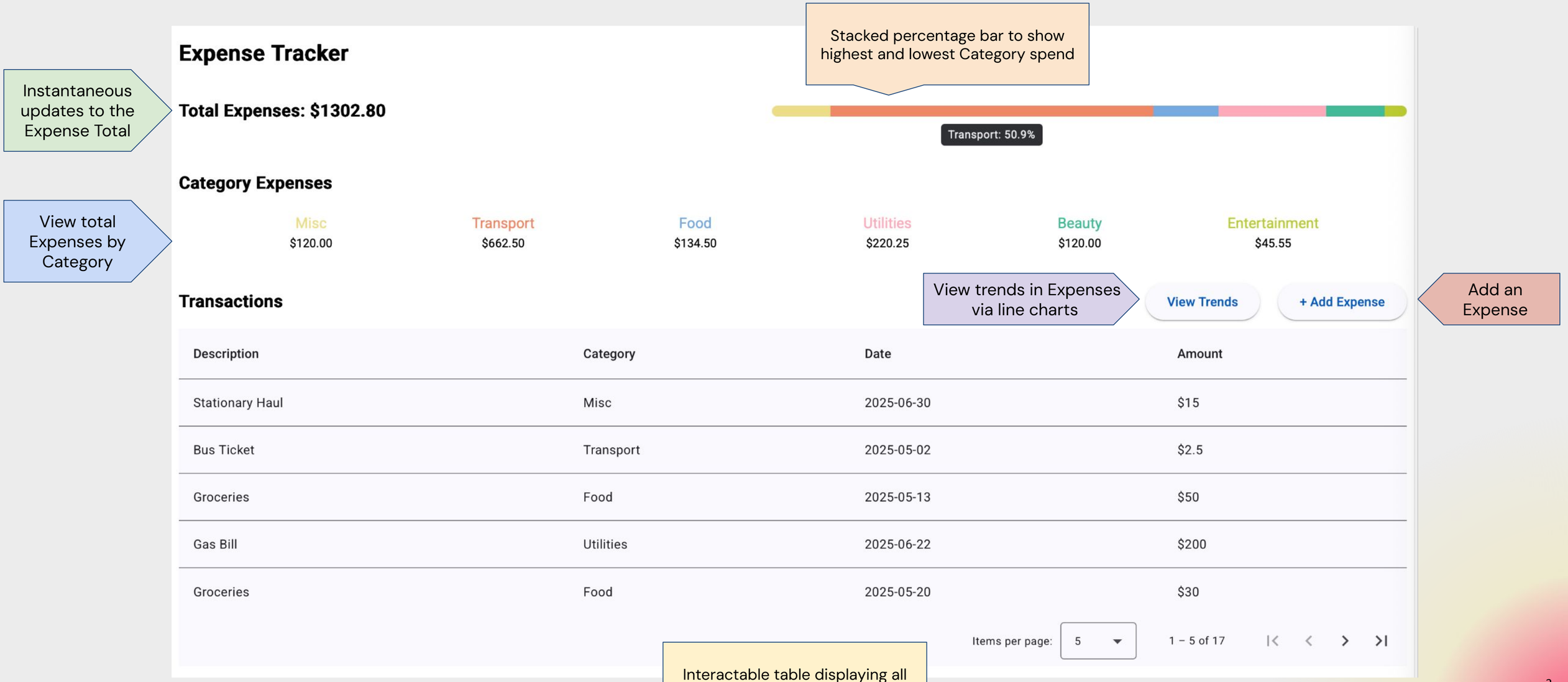
Sathvika Iyengar

Overview and Approach

- Goal: Create a simple Expense Tracker with an interactive UI and organized data management
- *For keeping track of expense data:* SQL Database
    - expenses.db (id, description, amount, category, date)
- *For managing queries between Frontend and Database:* Python/Flask
- *For a dynamic and scalable UI:* Angular



App Design



Stacked percentage bar to show highest and lowest Category spend

Instantaneous updates to the Expense Total

View total Expenses by Category

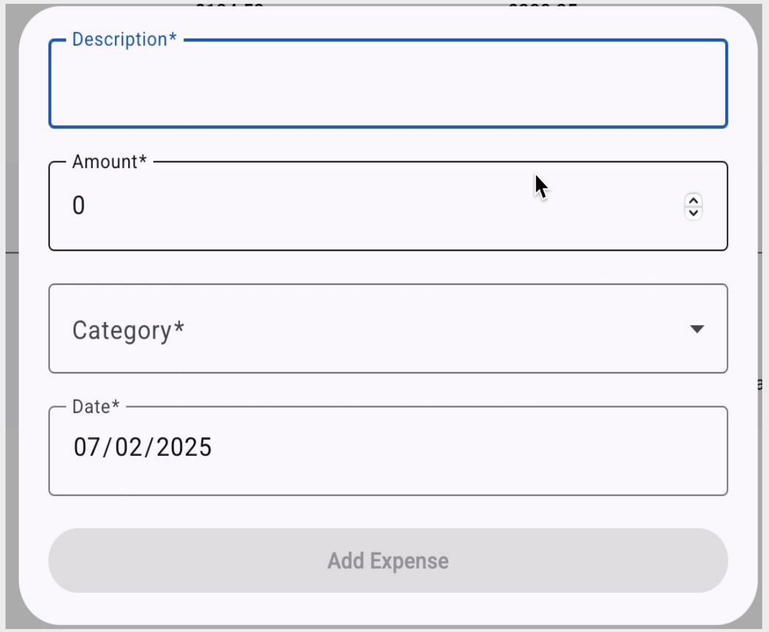
View trends in Expenses via line charts

Add an Expense

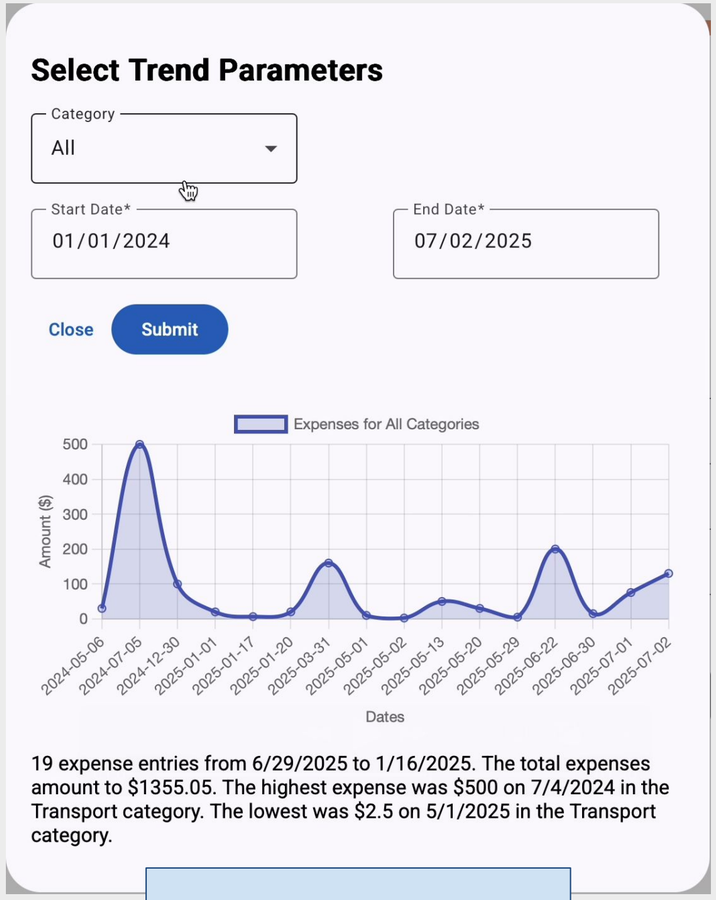
Interactable table displaying all Expenses

Feature Popups

Add an Expense by providing a description, amount, category, and date



View trends for all Expenses over a specific time range

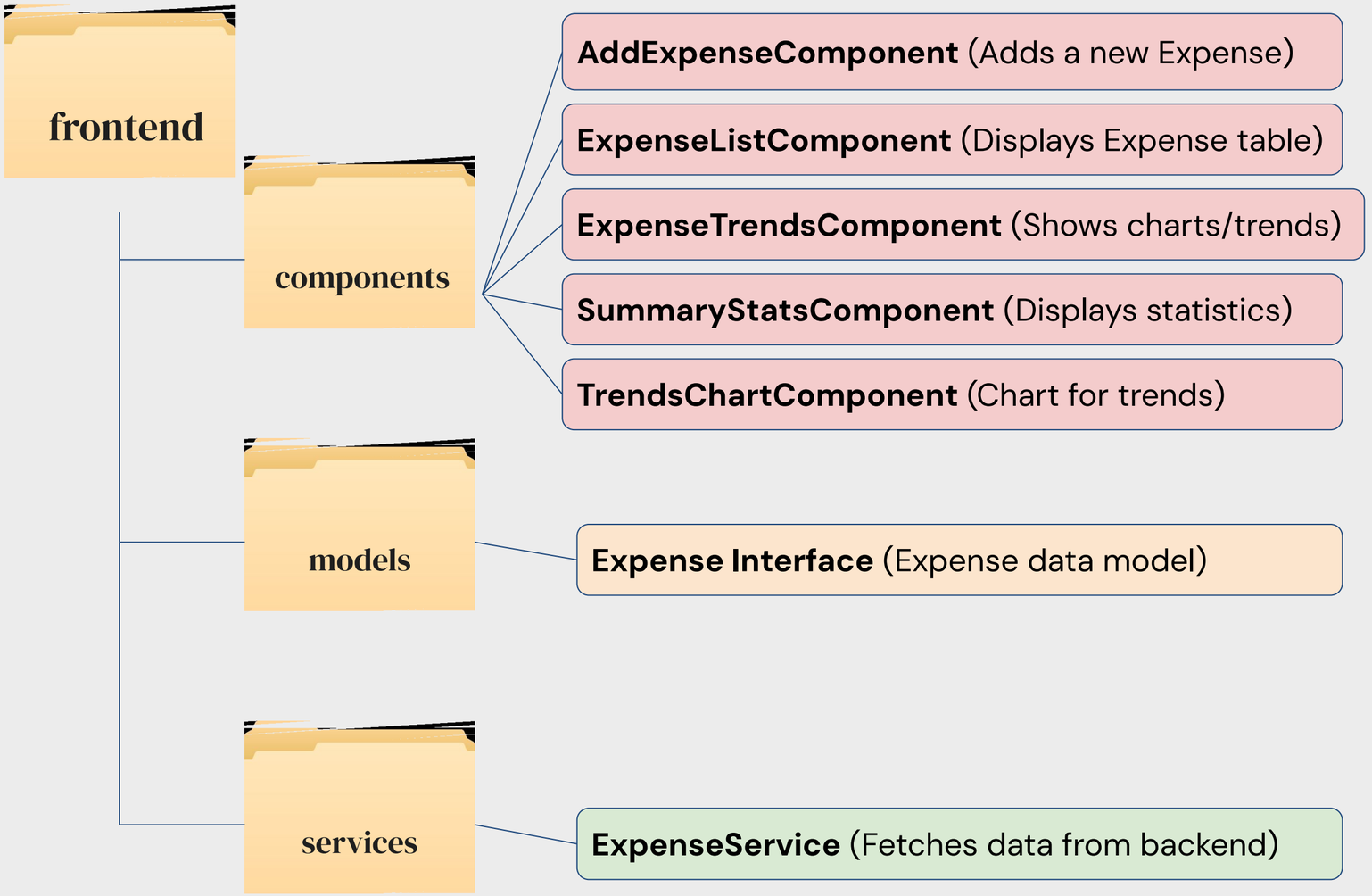


Short summaries provide about selected data

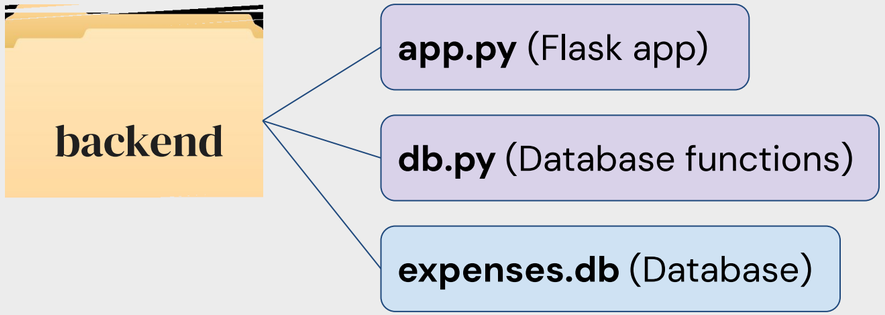
Or view trends for a specific category and time range

Key Folders and Files

Frontend



Backend



Installation/Run Directions

This application consists of an Angular frontend a Python/Flask backend to be run in separate terminals.

To run the ***Angular front end***:

> cd frontend/expense-tracker

> npm install

> cd src/app

> ng serve

To run the ***Python backend***:

> cd backend

> python app.py

Some package installations may be required via > pip install -r requirements.txt

For virtual environment:

> python3 -m venv env

> source env/bin/activate

> python app.py

This information is also available in the project's README.

Demo of Expense Tracker

[GitHub Repository]

Next Steps

- Advanced table operations (Delete, Update, Sort, Filter)
- NGRX state management (Actions, Reducers, Effects)
- Natural Language querying of data
- Custom data upload