Hey Presto Screenshots
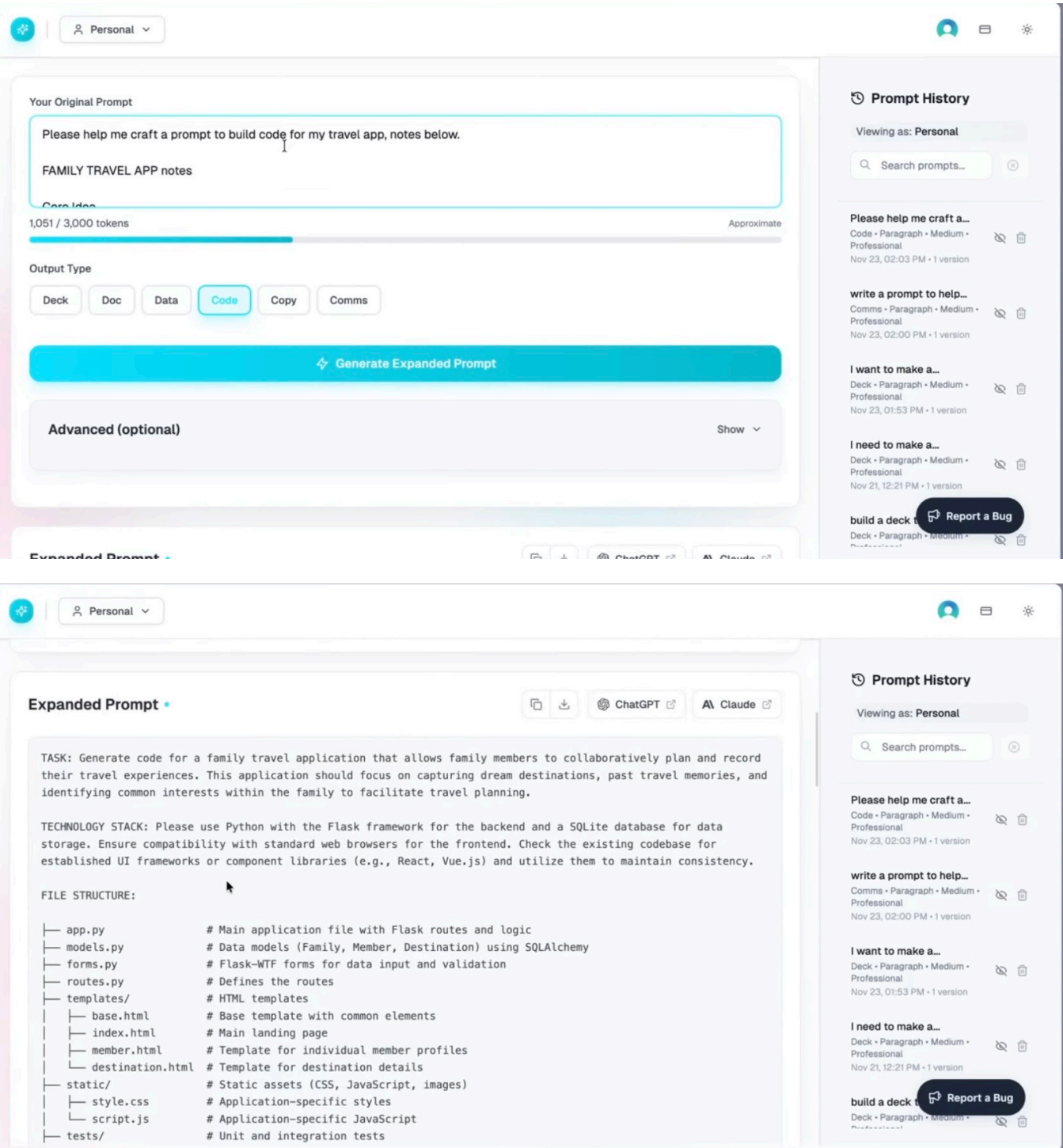
TASK: Generate code for a family travel application that allows family members to collaboratively plan and record their travel experiences. This application should focus on capturing dream destinations, past travel memories, and identifying common interests within the family to facilitate travel planning.

TECHNOLOGY STACK: Please use Python with the Flask framework for the backend and a SQLite database for data storage. Ensure compatibility with standard web browsers for the frontend. Check the existing codebase for established UI frameworks or component libraries (e.g., React, Vue.js) and utilize them to maintain consistency.

FILE STRUCTURE:

```
├── app.py              # Main application file with Flask routes and logic
├── models.py           # Data models (Family, Member, Destination) using SQLAlchemy
├── forms.py            # Flask-WTF forms for data input and validation
├── routes.py           # Defines the routes
├── templates/          # HTML templates
│   ├── base.html       # Base template with common elements
│   ├── index.html      # Main landing page
│   ├── member.html     # Template for individual member profiles
│   └── destination.html # Template for destination details
├── static/             # Static assets (CSS, JavaScript, images)
│   ├── style.css       # Application-specific styles
│   └── script.js       # Application-specific JavaScript
├── tests/              # Unit and integration tests
│   ├── conftest.py     # Pytest configuration file
│   ├── test_models.py  # Tests for data models
│   └── test_routes.py  # Tests for Flask routes
├── README.md           # Project documentation
└── requirements.txt    # Project dependencies
```

Report a Bug

DATA MODELS (models.py):

```
class Family:
    family_id: int (primary key, autoincrement)
    family_name: str (e.g., "The Smiths")
    members: list of Members (one-to-many relationship)

class Member:
    member_id: int (primary key, autoincrement)
    family_id: int (foreign key to Family)
    name: str
    dream_destinations: list of Destinations (many-to-many relationship)
    visited_destinations: list of Destinations (many-to-many relationship)
    favorite_climate: str [Illustrative: "Tropical", "Temperate", "Arctic"]
    favorite_activities: str [Illustrative: "Hiking", "Museums", "Beaches"]
    energy_level: str [Illustrative: "High", "Medium", "Low"]
    travel_style: str [Illustrative: "Planner", "Wanderer"]

class Destination:
    destination_id: int (primary key, autoincrement)
    name: str
    country: str
    kid_notes: str (notes from kids about the destination)
    adult_notes: str (notes from adults about the destination)
    tags: str [Illustrative: "Food", "Nature", "Animals", "Art"]
```

FORMS (forms.py): Use Flask-WTF for form handling.

  • MemberForm: Fields for name, favorite_climate, favorite_activities, energy_level, and travel_style.

Report a Bug