



Zen

AI Habit Tracker

Problem Statement:

Users face challenges in managing their daily routines, as existing tools often lack integration and personalization in tracking habits, goals, and productivity.

Our Solution:

Our project provides a comprehensive solution with:

- A Habit Tracker that seamlessly integrates habit and goal tracking.
- A Note Taking Module and an AI Chatbot for efficient task management.
- A Focus Mode that combines a Pomodoro timer with Zen Garden visualization to track progress.

Zen Garden:

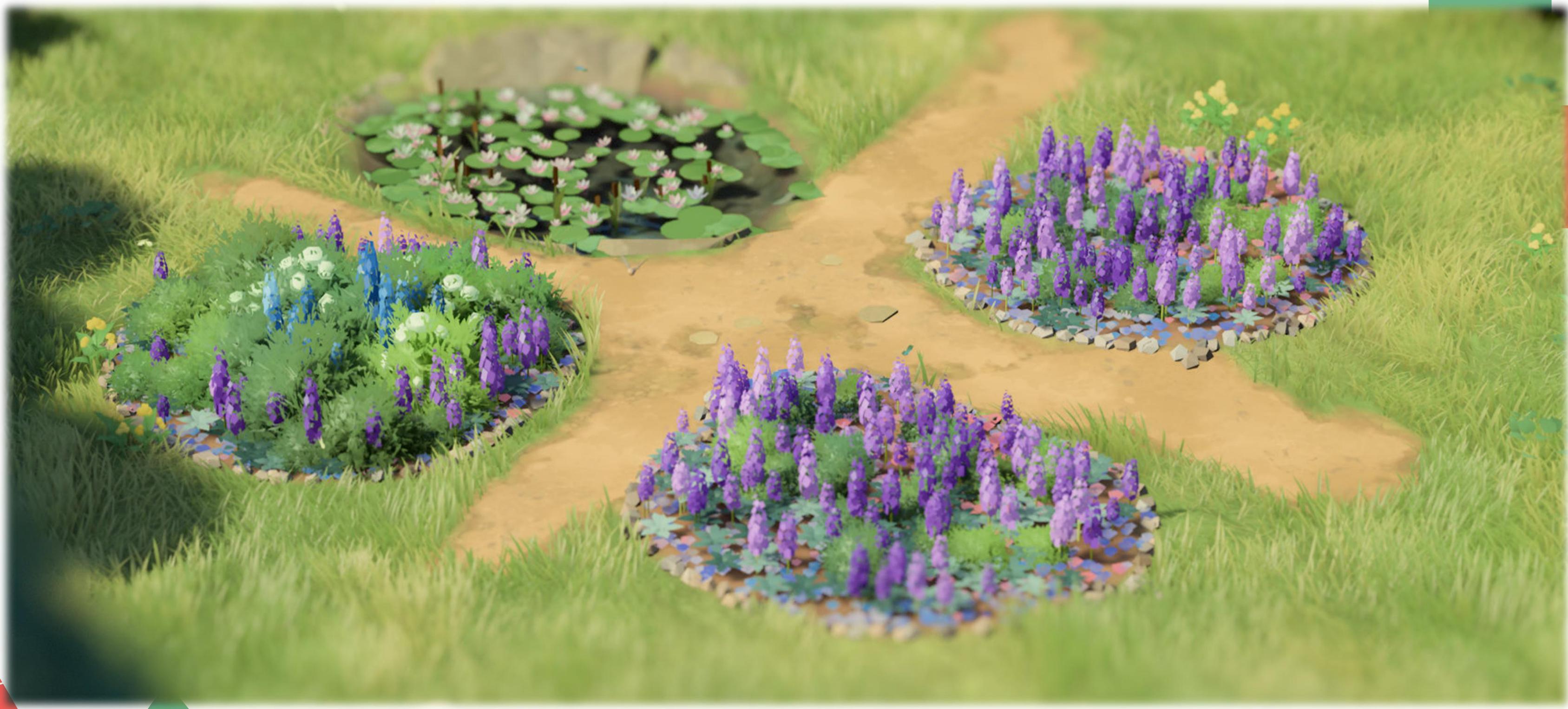
What it is:

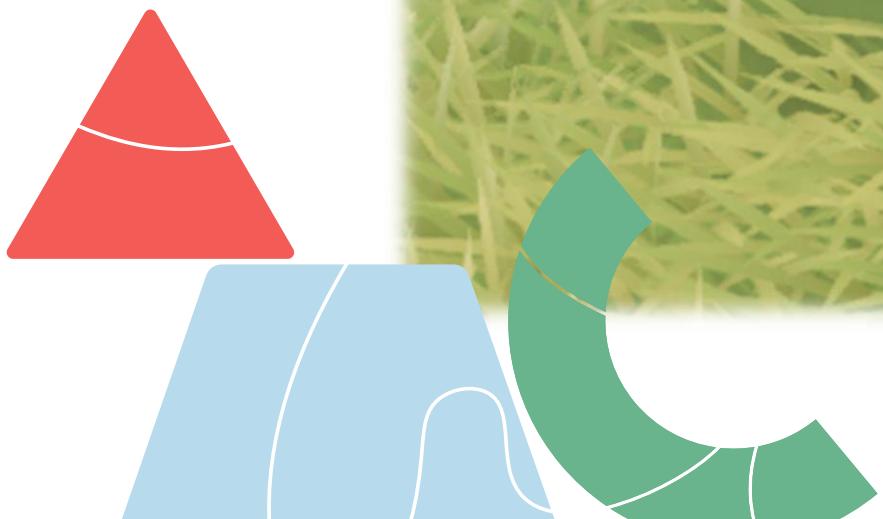
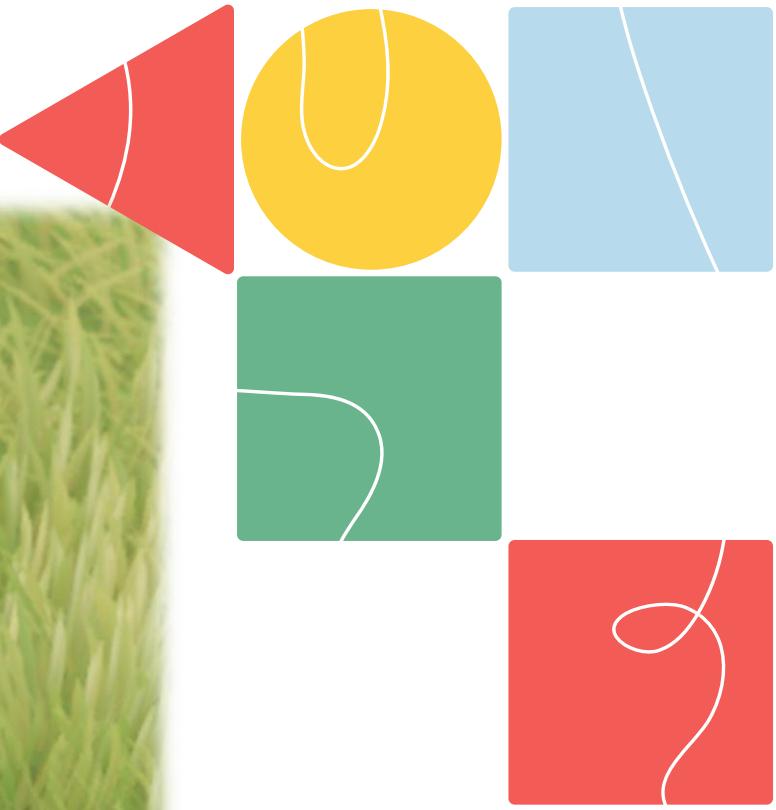
- Inbuilt gamification feature to retain user interest.
- Visual representation of the user's productivity and their progress towards their goals

How it works:

- Personalised goals, with each goal having a set of habits that the user must complete in order to achieve that goal
$$G = \{H_1, H_2, H_3, \dots, H_n\}$$
- Each goal gets a patch of land in the garden, with a unique flower for each habit in the patch.
- More consistent → healthier and taller the flower
- 2 properties of each flower: length and colour

(This is a preliminary representation of how the feature will look)



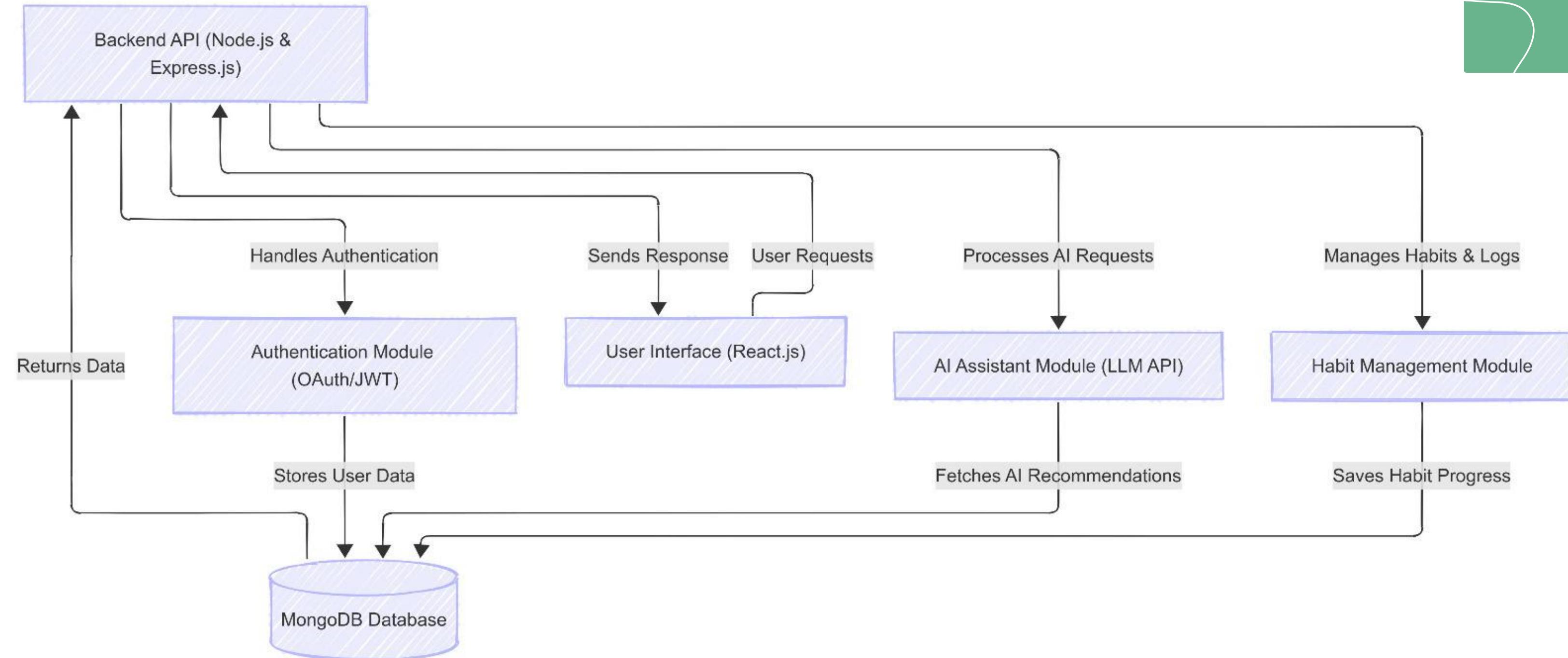


Roles and Responsibilities:

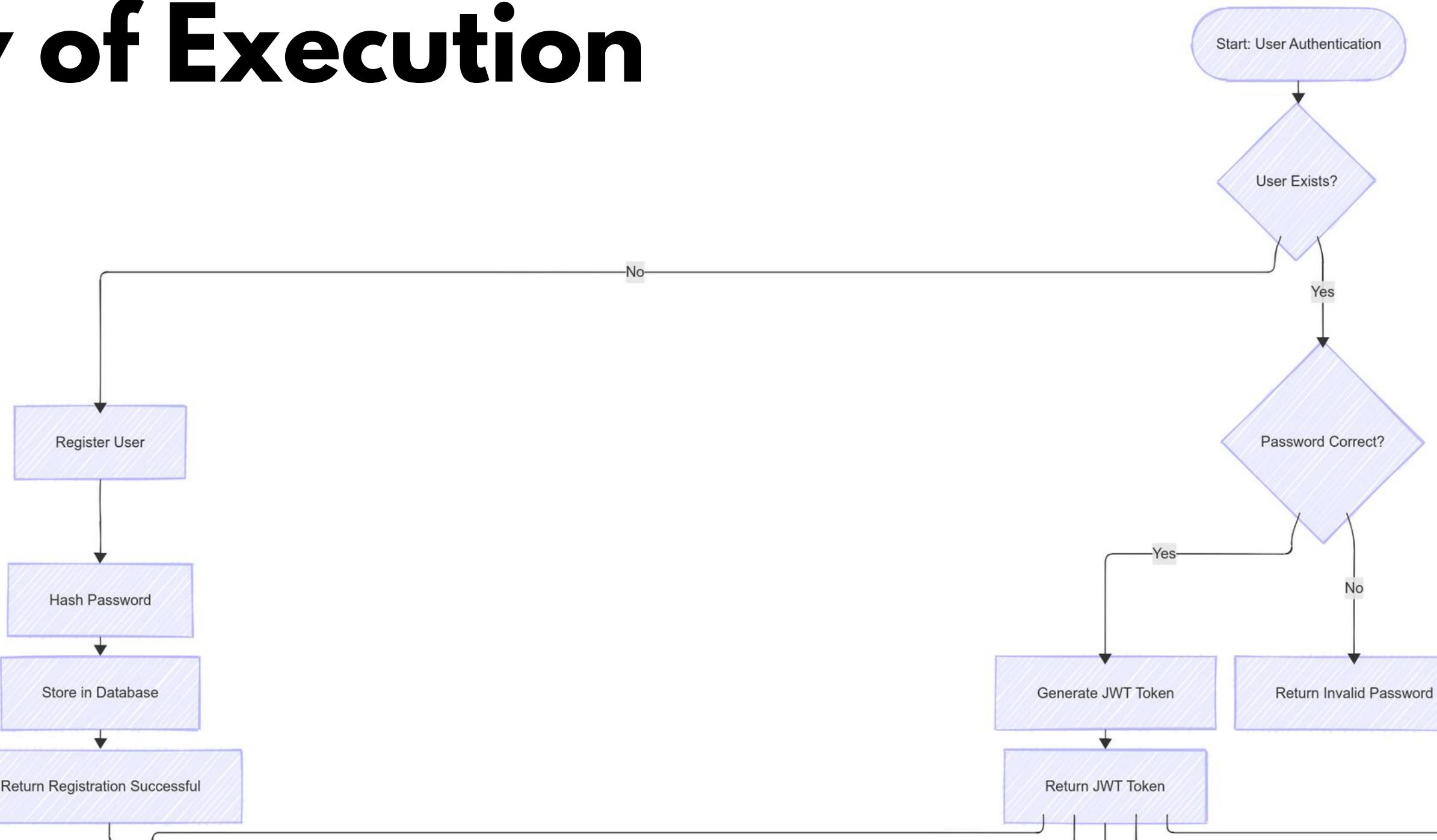
- Srachet Rai - Security & Front End
- Adarsh Bajpai - Front End
- Avijith Manikandan - AI Chat Bot
- Anmol Ranjan - Back End
- Abhi Bhardwaj - Database

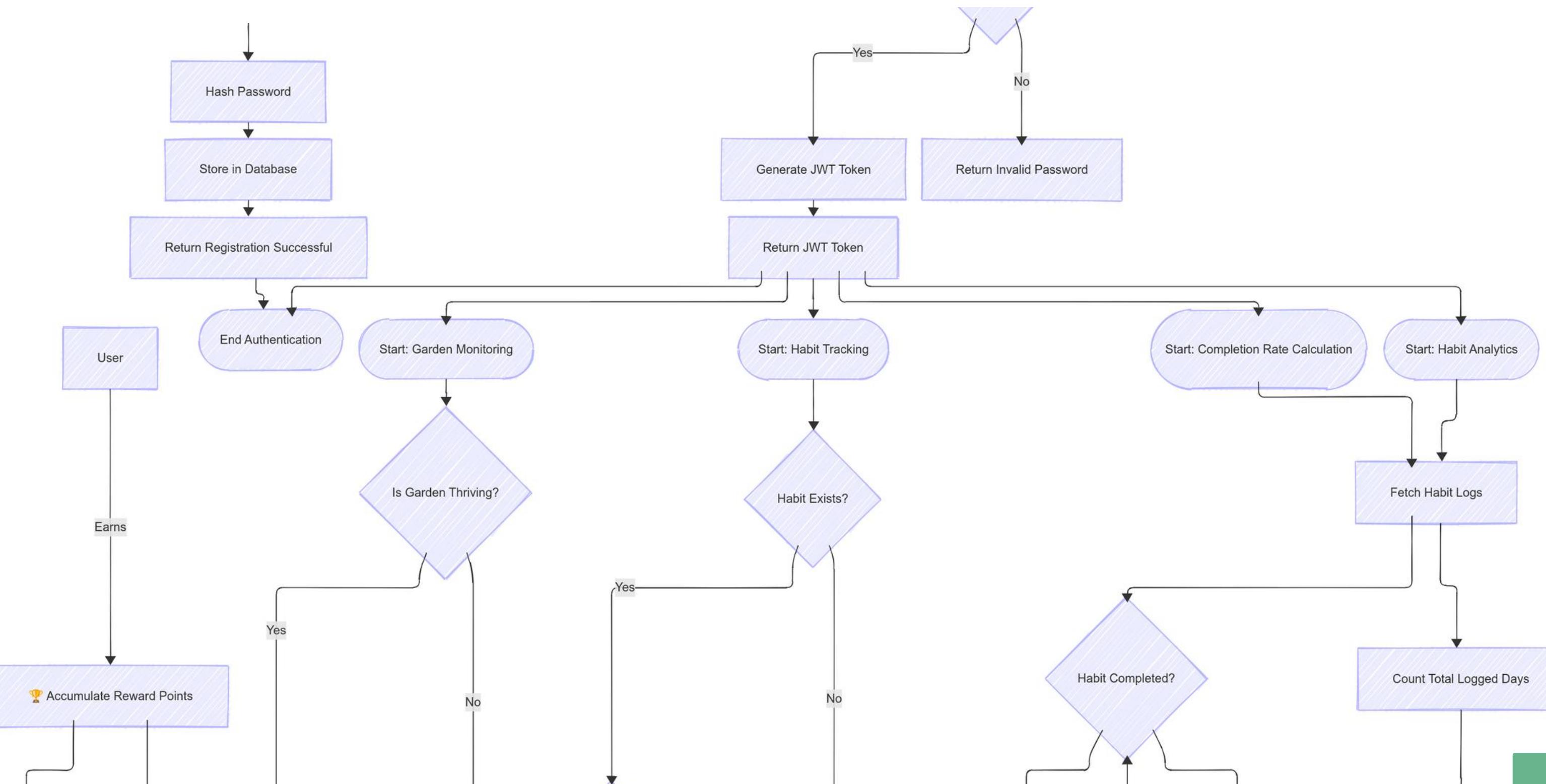
Tech Stack:

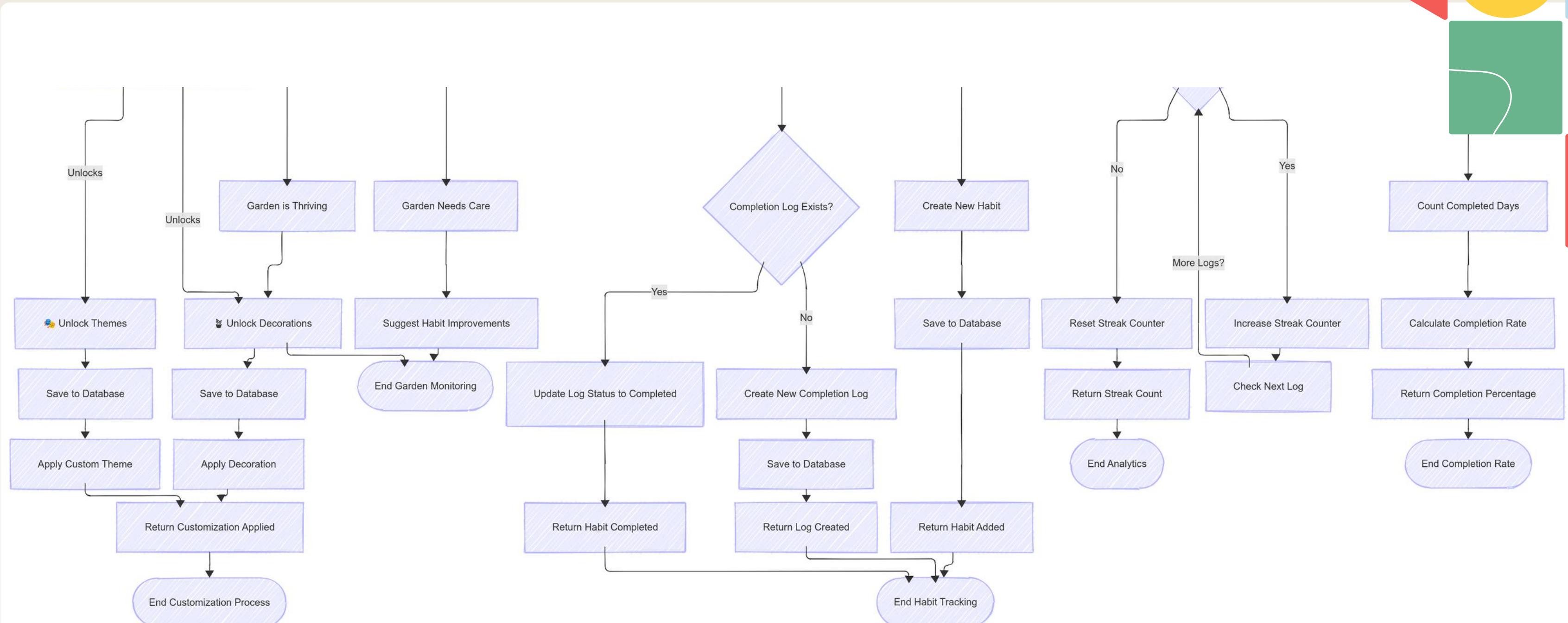
- Frontend (React.js) interacts with the Backend API.
- Backend API (Node.js & Express.js) manages:
 - Authentication Module (OAuth/JWT) for user logins.
 - AI Assistant Module (LLM API) for habit recommendations.
 - Habit Management Module for tracking user habits.
- Database (MongoDB) stores users, habits, logs, and AI recommendations.



Flow of Execution

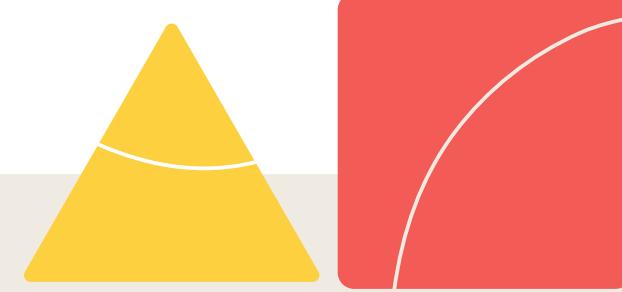
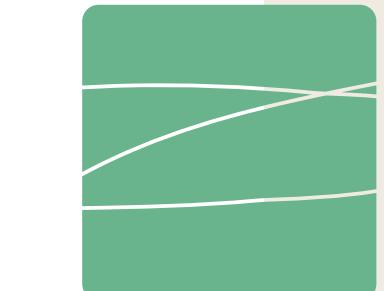
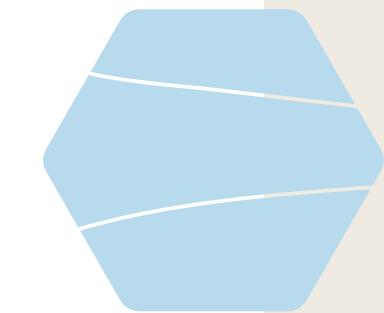
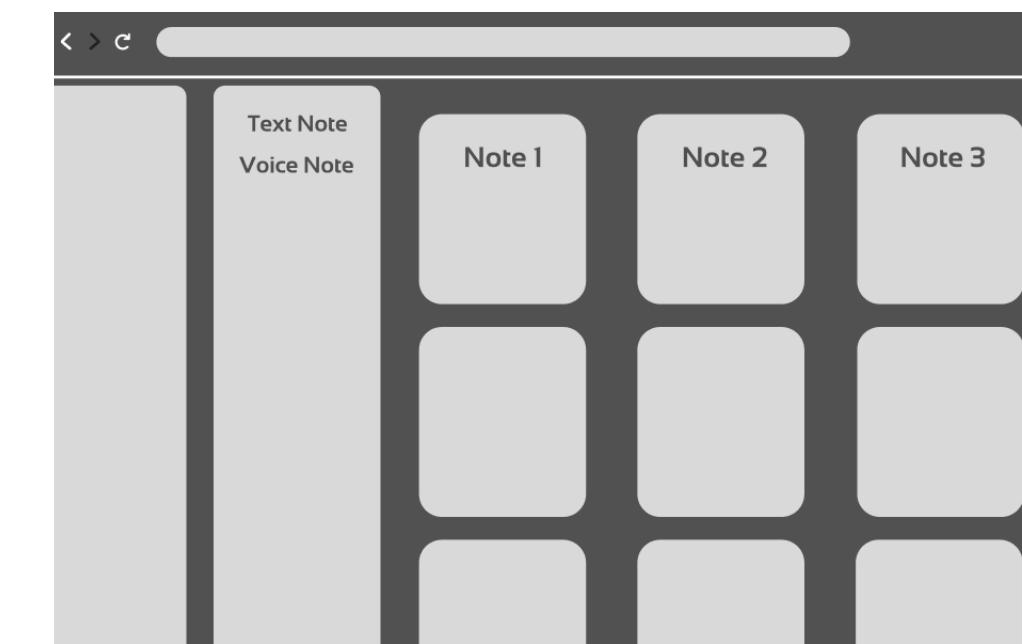
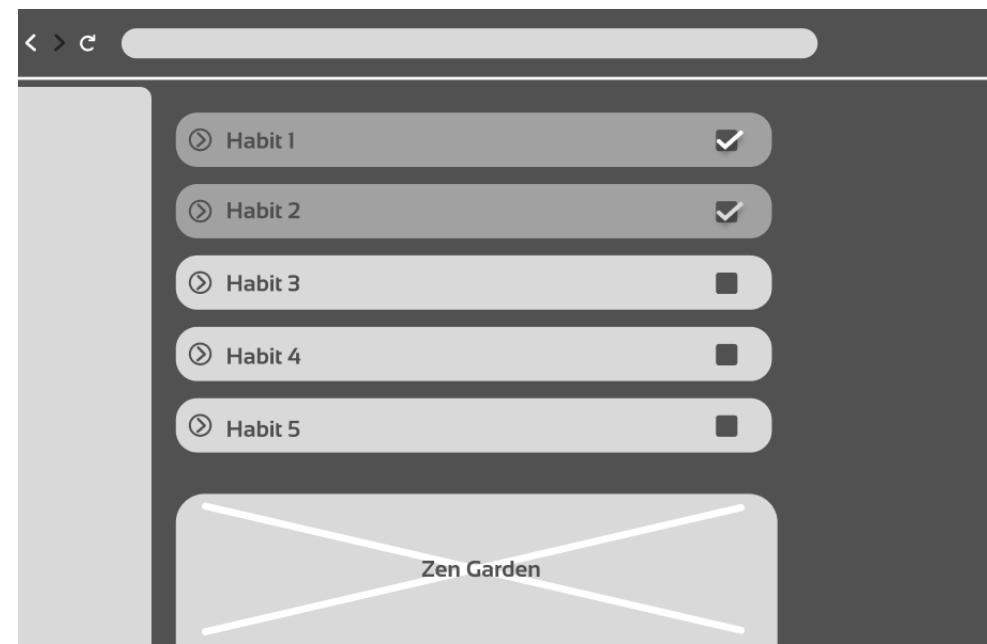
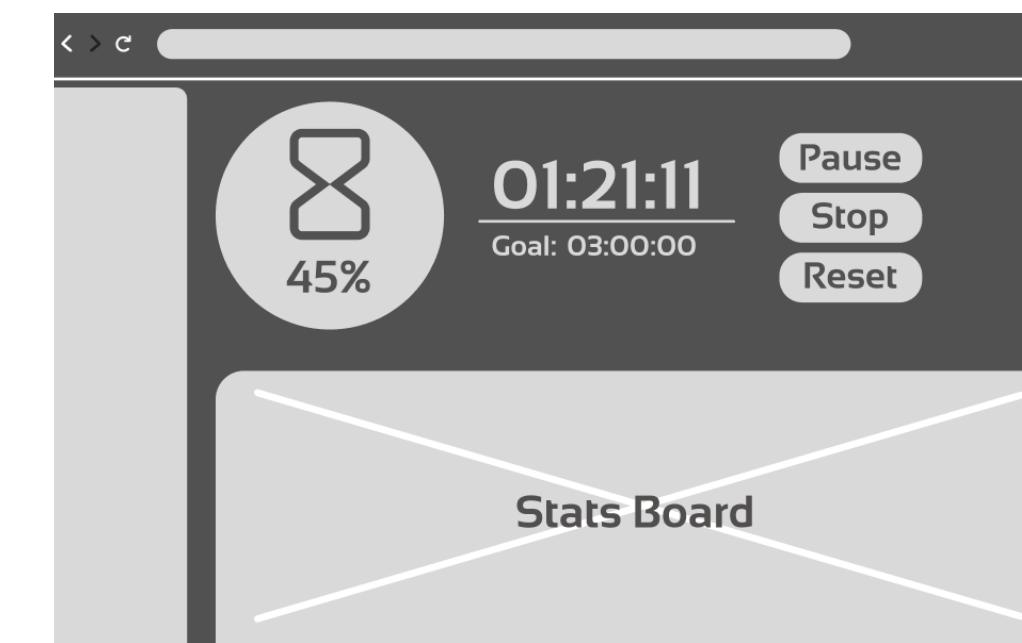
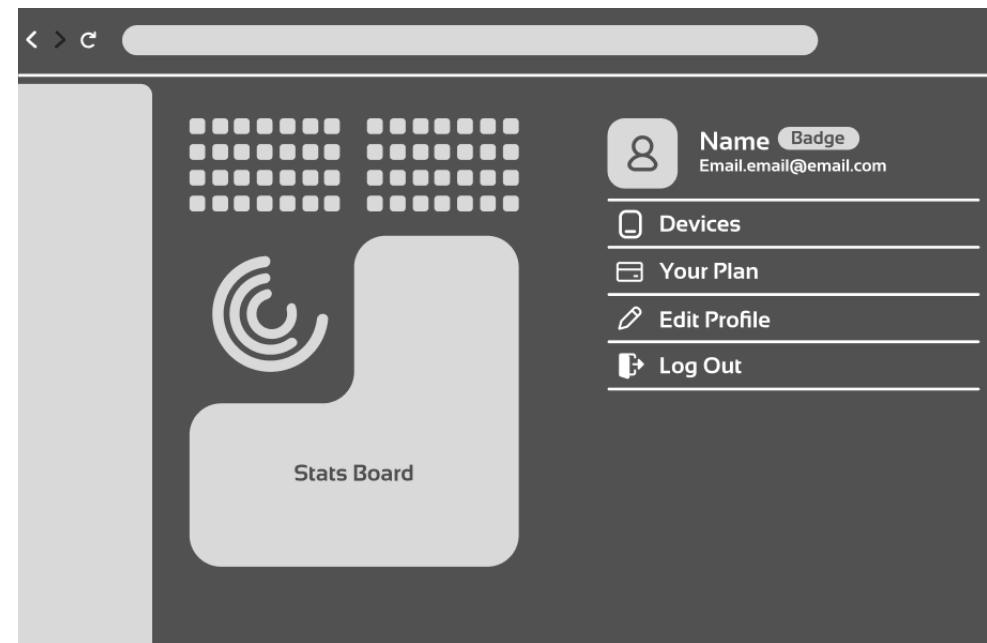




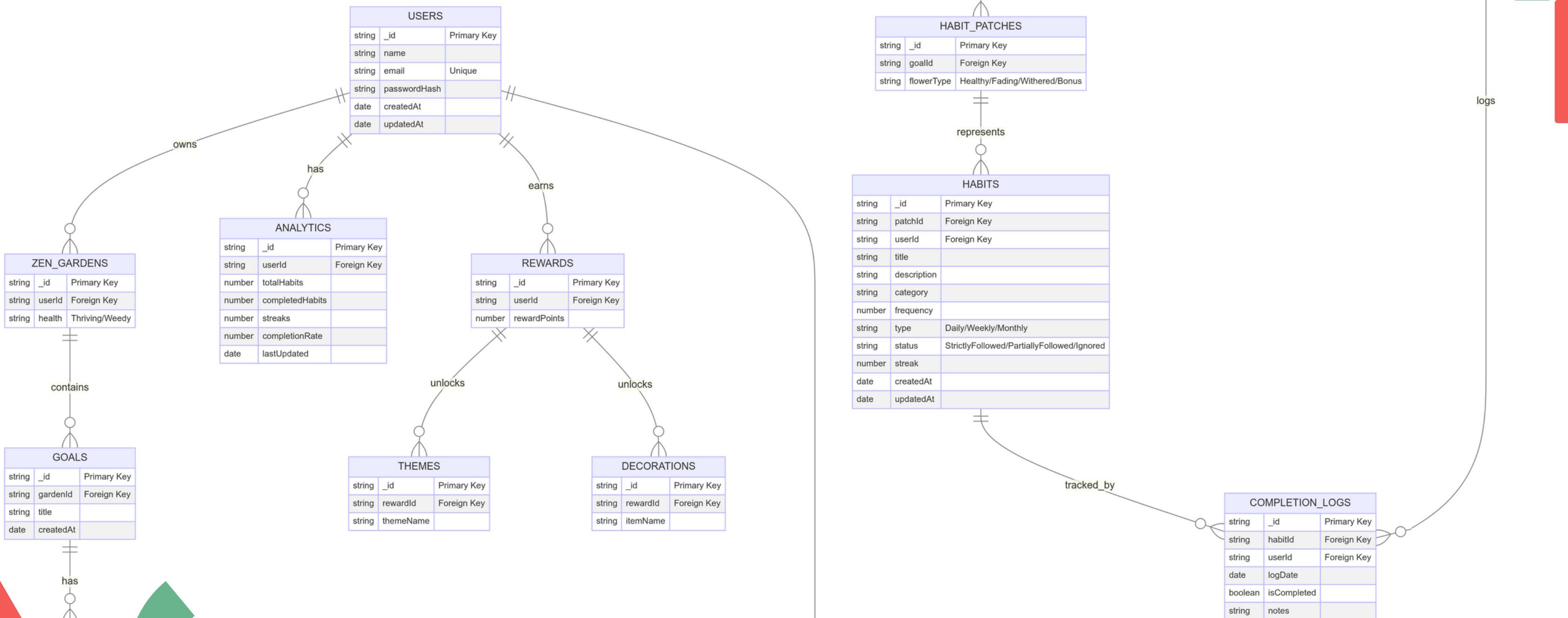


UI Wireframe/ Mockups:

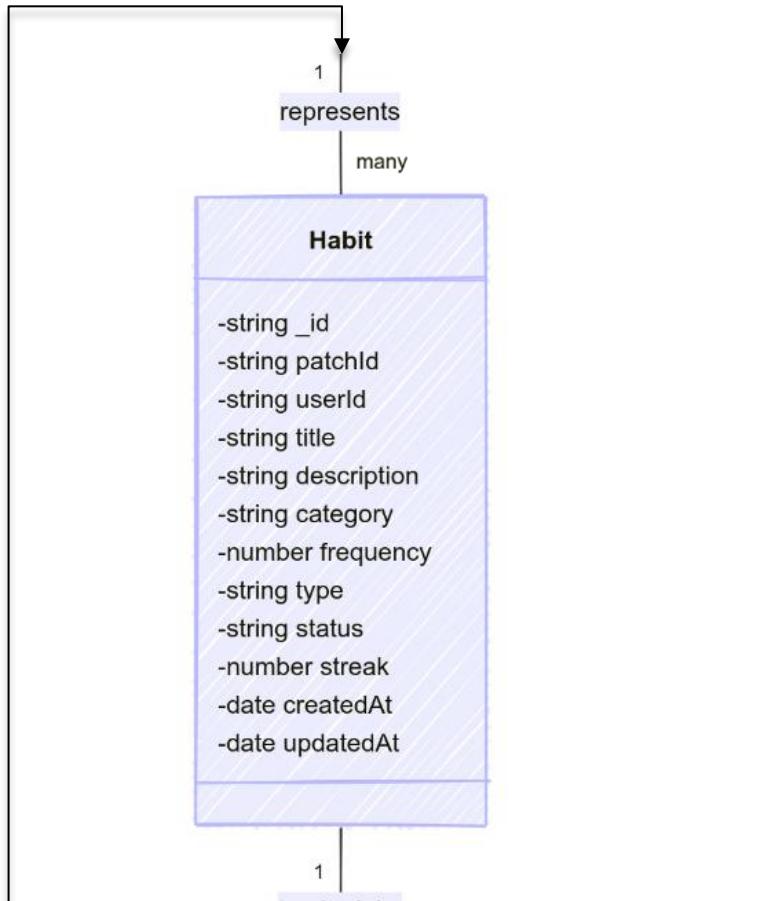
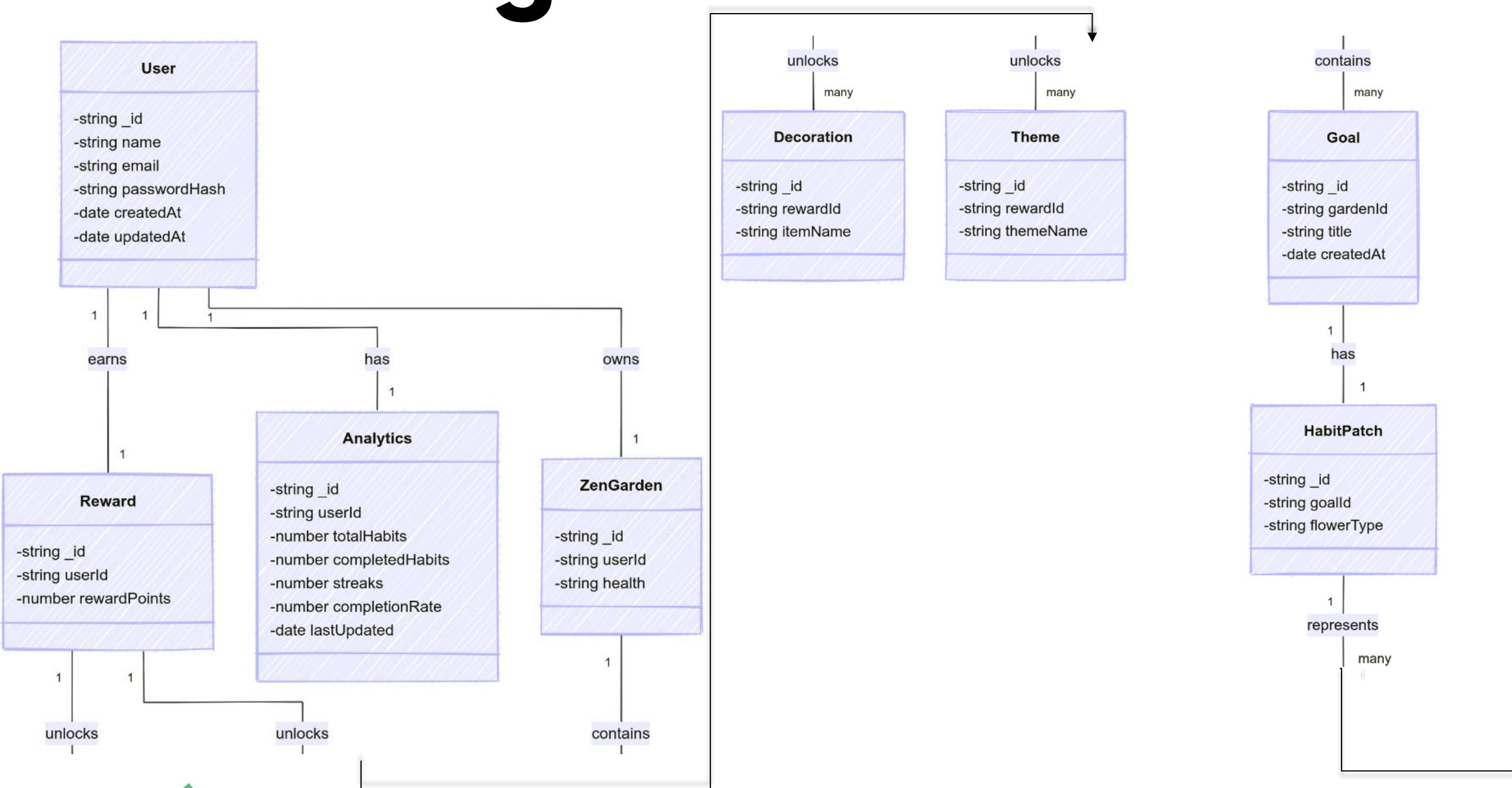
This is the basic UI workflow we developed to visualize the final layout of the web application.



ER Diagram:



Class Diagram:



Progress:

Task Done	Done By
Project Discussion	1st Feb
Finalizing Requirements	9th Feb
Functionality Discussion	12th Feb
SRS and SDD	19th Feb
Presentation	22nd Feb

Timeline:

Phase	Task/Activity	Date	Key Deliverables
Requirements & Planning	Finalize requirements, SDD, diagrams (ER, class), and requirements matrix.	Feb 24, 2025	Complete SDD and documentation, finalized requirements matrix, ER & class diagrams.
System & UI Design	Create system architecture, UI wireframes, and design mockups (focusing on key UI buttons and interactions).	Mar 5, 2025	Architecture diagrams and UI design sketches/mockups.
Database & Backend Development	Develop the database schema and implement core backend functionality (user auth, CRUD for goals/habits, etc.).	Mar 18, 2025	Functional database schema and backend API endpoints for core operations.
Frontend Development	Build and integrate UI components with backend services; implement main UI buttons and interactions.	Mar 25, 2025	Working user interface with key screens (authentication, dashboard, habit tracking, customization).
Integration & Testing	Integrate frontend with backend; perform unit and integration testing; fix bugs.	Apr 15, 2025	A stable, integrated system with test results and bug fixes documented.
Final Report & Documentation	Prepare final project report, update documentation, and compile any demonstration materials (screenshots, videos).	Apr 28, 2025	Final report, updated SDD, and all supporting documentation ready for submission.
Buffer & Final Adjustments	Last-minute tweaks and review to ensure all requirements are met and the final report is complete.	May , 2025	Final polished project and documentation; project ready for academic submission.



Thank You

Abhi Bhardwaj

Adarsh Bajpai

Avijith Manikandan

Anmol Ranjan

Srachet Rai