Web Programming: Project Proposal

Contents

1 Introduction

1.1 Project Information

Project Name: The Existential Choreography Planner: Mapping the Micro-Traumas of Daily Tasks

Supervisor: Farzana Tabassum

1.2 Team Details

Abrar Mahmud Hasan - 220041119
Nakib Saleh - 220041220
Md. Sazidur Rahman - 220041226

2 Project Overview & Motivation

2.1 Overview

The Existential Choreography Planner is a MERN-stack web application designed to fulfill the "Life, But Make It Weird" project theme. It takes simple, mundane household or academic chores and transforms them into hilariously dramatic, over-engineered "Choreography Flowcharts". The core function relies on visual representation to map the emotional, cognitive, and physical dependencies required to complete a trivial task, turning a to-do list into a complex process map.

2.2 Motivation behind the project

The primary motivation is to meet all technical requirements of the Web Programming Lab course while delivering a creative and novel solution. This project requires extensive use of modern React features (hooks, routing, conditional rendering) and demonstrates a robust backend including CRUD operations, multiple schemas, token-based authentication, OAuth, and file handling. The project emphasizes a complex, custom frontend design created from scratch, avoiding pre-made templates.

3 Key Features

- **Task Input & Existential Mapping:** Users input a simple daily chore (e.g., ”Mop the floor”). The system processes this input to automatically generate an intricate, multi-stage Choreography Flowchart, visually mapping the task’s exaggerated emotional and physical requirements

- **Custom Flowchart Interaction:** Users can directly manipulate the generated flowchart using the custom interface. They can drag nodes to rearrange the ”emotional dependencies,” rename steps to reflect personal angst, and re-route paths, saving the resulting map to the database.

- **Conditional Rendering of Decision Nodes:** The system includes custom Decision Nodes (e.g., ”Do I have the mental energy?”) that require complex conditional rendering logic to determine the path the user should take, demonstrating effective use of React features.

- **Token-Based Authentication & OAuth:** Users must register and sign in to access and manage their private ”Life Load” and archived flowcharts. Both token-based authentication and OAuth (e.g., Google Sign-In) will be implemented to secure user data.

- **Archiving the Trauma (File Handling):** Users can archive their completed flowcharts and upload an optional ”Failure Log” (a document or image explaining why the task was not completed successfully). This feature will implement robust file handling on the backend.

- **Multiple Schema Dependency:** The backend will manage at least three distinct schemas: UserSchema, ChoreSchema (for the input task), and FlowSchema (to store the intricate node, edge, and position data).

4 Team Roles

Abrar Mahmud Hasan	Frontend Development (React Components, Hooks, State Management)
Nakib Saleh	Backend development (Node.js, Express, Schemas, Authentication)
Md Sazidur Rahman	Database Management (MongoDB, File Handling, Middleware, Flow Integration)

5 Tools & Technologies

Category	**Tools**
Frontend	React, React Router, Custom CSS/SCSS
Backend	Node.js, Express.js, Monggose (MongoDB ORM)
Database	MongoDB (NoSQL)
Version Control	Git & Github
Authentication	JSON Web Tokens(JWT), OAuth

6 Project Timeline

Week	**Area of Focus**
1-2	Git setup and initial commits. Basic MERN stack configured. User registration and login (Token Auth implemented). ChoreSchema and UserSchema defined.
3-4	Initial visual canvas setup. Custom node/edge design. Basic CRUD operations on ChoreSchema. FlowSchema defined.
5-6	Logic for automatic flow generation from simple text input. Implementation of custom Decision Nodes (conditional rendering). User interaction: saving custom node positions and connections.
7-8	OAuth implementation. File handling for "Failure Log" uploads. Final middleware implementation and robust error handling. Final design polish and responsiveness check.