Title

Meme Generator Application

Description

This project aims to develop a Meme Generator App that allows users to create, customize, and share memes easily. Users can select images, add text, and generate memes that can be saved or shared on social platforms.

Scope

- Target Audience: Social media users, meme enthusiasts, content creators.
- Content: Users can choose from a library of images or upload their own, add customizable text, and generate memes.
- Formats: The app will be available as a web application.
- Features:
    - Image selection from a predefined library.
    - Text customization (font, size, color).
    - Save and share options for generated memes.
-

Objectives

- User Engagement: Provide an easy-to-use interface for meme creation to enhance user interaction.
- Social Sharing: Facilitate sharing of created memes across various social media platforms.
- Brand Growth: Establish the app as a go-to tool for meme generation.

Architecture Details

- Architecture Pattern: Monolithic architecture.
- Containerization: Single container with a single component that encapsulates the entire application (frontend and backend).

Monolithic Architecture Overview

In a monolithic architecture, all components of the application (UI, business logic, data access) are integrated into a single codebase. This simplifies deployment and scaling but may pose challenges in flexibility and maintainability as the app grows.

Technology Details

- Frontend Technologies: HTML, CSS, JavaScript (React or Vue.js).
- Backend Technologies: Node.js with Express for server-side logic.
- Database: MongoDB or PostgreSQL for storing user data and meme metadata.
- Image Processing Library: Use libraries like `html2canvas` for capturing meme images.
- Hosting: Deploy on platforms like Heroku or AWS.

Team Composition

- Project Manager: Oversees project timelines and team coordination.
- Frontend Developer(s): Responsible for UI/UX design and implementation.
- Backend Developer(s): Manages server-side logic and database interactions.
- Designer: Creates assets and ensures the app has an appealing visual design.
- QA Tester: Conducts testing to ensure functionality and user experience are optimal.

Timeline Details

1. Week 1 (Jan 28 - Feb 3): Define project goals, scope, and assemble the team; finalize tech stack.
2. Week 2 (Feb 4 - Feb 10): Design UI/UX mockups; begin frontend development.
3. Week 3 (Feb 11 - Feb 17): Develop backend services; set up database schema and API endpoints.
4. Week 4 (Feb 18 - Feb 24): Integrate frontend with backend; implement image processing features.
5. Week 5 (Feb 25 - Mar 3): Conduct thorough testing; fix bugs and optimize performance.
6. Week 6 (Mar 4 - Mar 10): Launch the app; initiate marketing strategies to promote user engagement.