

Title

Meme Gen. App

Description

This proj. aims to dev. a Meme Gen. App that allows users to create, customize, and share memes easily. Users can select imgs, add text, and gen. memes that can be saved or shared on social media (SM).

Scope

- Target Aud.: SM users, meme enthusiasts, content creators.
- Content: Users can choose from a library of imgs or upload their own, add customizable text, and gen. memes.
- Formats: The app will be available as a web app (WA).
- Features:
 - Img selection from a predefined library.
 - Text customization (font, size, color).
 - Save and share options for gen. memes.
-

Objectives

- User Eng.: Provide an easy-to-use UI for meme creation to enhance user interaction.
- Social Sharing: Facilitate sharing of created memes across various SM platforms.
- Brand Growth: Establish the app as a go-to tool for meme gen.

Architecture Details

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

Monolithic Architecture Overview

In a monolithic architecture, all components of the app (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 Techs: HTML, CSS, JS (React or Vue.js).
- Backend Techs: Node.js with Express for server-side logic.
- DB: MongoDB or PostgreSQL for storing user data and meme metadata.
- Img Processing Lib: Use libs like `html2canvas` for capturing meme imgs.
- Hosting: Deploy on platforms like Heroku or AWS.

Team Composition

- PM: Project Manager oversees proj. timelines and team coordination.
- FE Dev(s): Frontend Developer(s) responsible for UI/UX design and implementation.
- BE Dev(s): Backend Developer(s) manages server-side logic and DB interactions.
- Designer: Creates assets and ensures the app has an appealing visual design.
- QA Tester: Conducts testing to ensure functionality and UX are optimal.

Timeline Details

1. Week 1 (Jan 28 - Feb 3): Define proj. goals, scope, and assemble the team; finalize tech stack.
 2. Week 2 (Feb 4 - Feb 10): Design UI/UX mockups; begin FE dev.
 3. Week 3 (Feb 11 - Feb 17): Develop BE services; set up DB schema and API endpoints.
 4. Week 4 (Feb 18 - Feb 24): Integrate FE with BE; implement img 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 mktg. strategies to promote user engagement.
- 1.