Meme Generator

OBJECTIVE:

- ✓ To create a web-based application that allows users to create, customize, and share memes.
- ✓ To provide a user-friendly interface where users can add pictures, select popular meme templates, add custom text, and generate memes.
- ✓ To download generated memes and share it directly on social media platforms. (using a generated link).

Components:

1. **Frontend:**
   - React.js .
   - HTML/CSS.
   - Axios for API calls.
2. **Backend:**
   - Node.js
   - Express.js
   - RESTful APIs
3. **Database:**
   - MongoDB
   - Mongoose
4. **Hosting Platform**:
   - Netlify /Vercel
   - Heroku
   - MongoDB Atlas

FRONTEND ARCHITECTURE:

Key components include:

1) **Login page:** Authenticates the user using their mail and creates an account for the user
2) **Home Page**: Displays popular meme templates and a search bar.
3) **Meme Editor**: Allows users to select a template, add pictures and text, and customize the meme.
4) **Preview Page**: Shows the final meme with options to download or share.
5) **User Dashboard:** Allows users to save and manage their memes.
6) **Responsive Design**: Ensures the application works seamlessly on desktop and mobile devices.

BACKEND ARCHITECTURE:

1) **Template Management**: Fetches meme templates from a database or external API.
2) **Meme Generation API**: Accepts user input (text, template ID) and generates a meme using a library sharp.
3) **User Management**: Handles user authentication and meme storage.
4) **Sharing API**: Integrates with social media platforms for sharing memes.

DATABASE ARCHITECTURE:

1) **Meme Templates**: Preloaded meme image templates
2) **User-Generated Memes**: retrieve memes created by users, linked to their accounts.

HOSTING:

1. **Frontend Hosting**:
    - **Netlify** or **Vercel** for static React apps.
2. **Backend Hosting**:
    - **Heroku** for Node.js applications.
3. **Database Hosting**:
    - **MongoDB Atlas** for MongoDB hosting.

FLOW OF WEBPAGE:

User Interaction Flow:

1. **User logins using the email-id:**
    - The user logins using their registered email id or creates an account if he/she is a new user.
2. **User Visits Home Page**:
    - The user lands on the homepage, which displays popular meme templates.
3. **User Selects a Template**:
    - The user clicks on a template, which redirects them to the meme editor.
4. **User Customizes Meme**:
    - The user adds text, adjusts font size, and customizes the meme.
5. **User Generates Meme**:
    - The user clicks "Generate," and the frontend sends a request to the backend.
6. **Backend Processes Request**:
    - The backend fetches the template, overlays the text, and generates the meme.

7. **Meme Preview**:
    - The generated meme is displayed on the preview page with options to download or share.
8. **User Shares Meme**:
    - The user shares the meme on social media or downloads it.

System Flow:

1. **Frontend**:
    - Handles user input and displays the UI.
    - Communicates with the backend via RESTful APIs.
2. **Backend**:
    - Receives requests from the frontend.
    - Processes meme generation and interacts with the database.
3. **Database**:
    - Stores meme templates and user-generated memes.
4. **Hosting Platform**:
    - Ensures the application is accessible globally.

FLOW DIAGRAM: