Batch: D-2	**Roll No: 16010122323**
Exp No: 3	

Title: Prepare the Design document and Specification of mini project. (Design Phase).

Objective: To prepare design document for implementation of mini project after appropriate analysis of gathered requirement.

Expected Outcome of Experiment:

Course Outcome	After successful completion of the course students should be able to
CO 3	Describe the design in the form of algorithm/flowchart/block diagram.

Books/ Journals/ Websites referred:
1.
2.
3.

Title of Mini Project: TO-DO list

Team Members: < To be filled in by the student>

1. Kartik Ambupe-16010122318
2. Jiya Trivedi 16010122321
3. Vedansh Savla 16010122323
4

“Students are required to prepare design document in the format given below, please replace the words shown in yellow background of the template.”

1. System Design

1. ***Task Component:***
   - *Display tasks with options to mark as completed, delete, or edit.*

- *Use state to manage the task list.*

2. ***Add Task Component:***
   - *Allow users to add new tasks.*
   - *Validate and sanitize input.*
3. ***Preferences Component:***
   - *Enable users to set task preferences (e.g., due dates, categories).*
4. ***Authentication Component:***
   - *Provide a login and signup interface.*
   - *Implement email/password authentication using Firebase.*
   - *Implement Google Sign-In using Firebase.*
5. ***App Component:***
   - *Manage the overall structure of the application.*
   - *Handle routing (react-router) for different components.*

Authentication (Firebase Authentication):

1. ***Email/Password Authentication:***
   - *Use Firebase Authentication to handle user signup and login with email/password.*
   - *Securely store and manage user credentials.*
2. ***Google Sign-In:***
   - *Integrate Firebase's Google Sign-In for a seamless authentication process.*
   - *Handle user authorization and profile management.*

Backend (Firebase Firestore):

1. ***Task Storage:***
   - *Use Firebase Firestore to store task data.*
   - *Organize tasks by user to maintain privacy.*
2. ***User Data:***
   - *Store user data such as preferences, completed tasks, etc.*

Integration:

1. ***Frontend-Backend Communication:***

- *Use Firebase SDK to communicate between frontend and backend.*
- *Implement CRUD operations for tasks.*

2. ***Real-Time Updates:***
   - *Leverage Firebase real-time features to update the UI when tasks are added, deleted, or modified.*

Security:

1. ***Authentication Security:***
   - *Implement proper Firebase security rules to restrict unauthorized access.*
2. ***Data Security:***
   - *Ensure that sensitive user data is properly secured.*
   - *Use HTTPS for secure communication.*

Deployment:

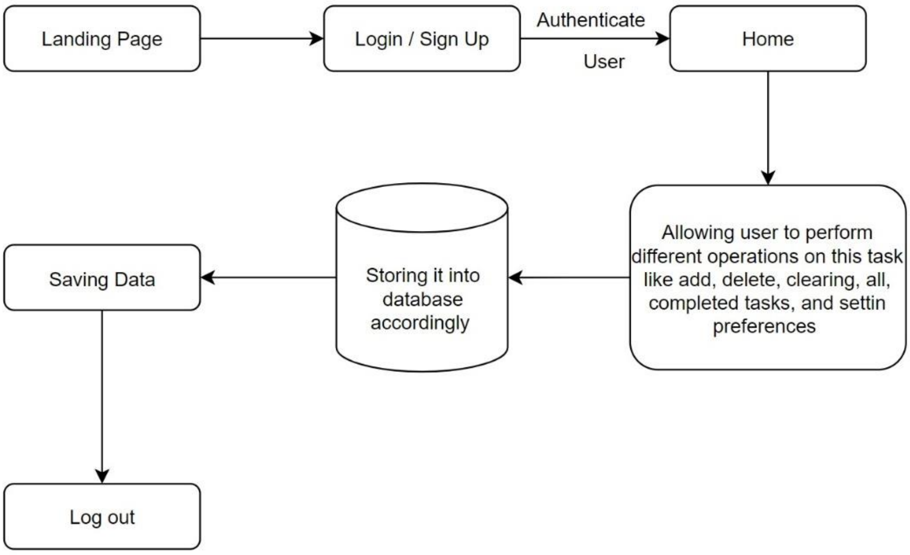
1. ***Hosting:***
   - *Deploy the React app using Firebase Hosting.*
2. ***Scalability:***
   - *Design the system to scale horizontally to accommodate a growing user base.*

Testing:

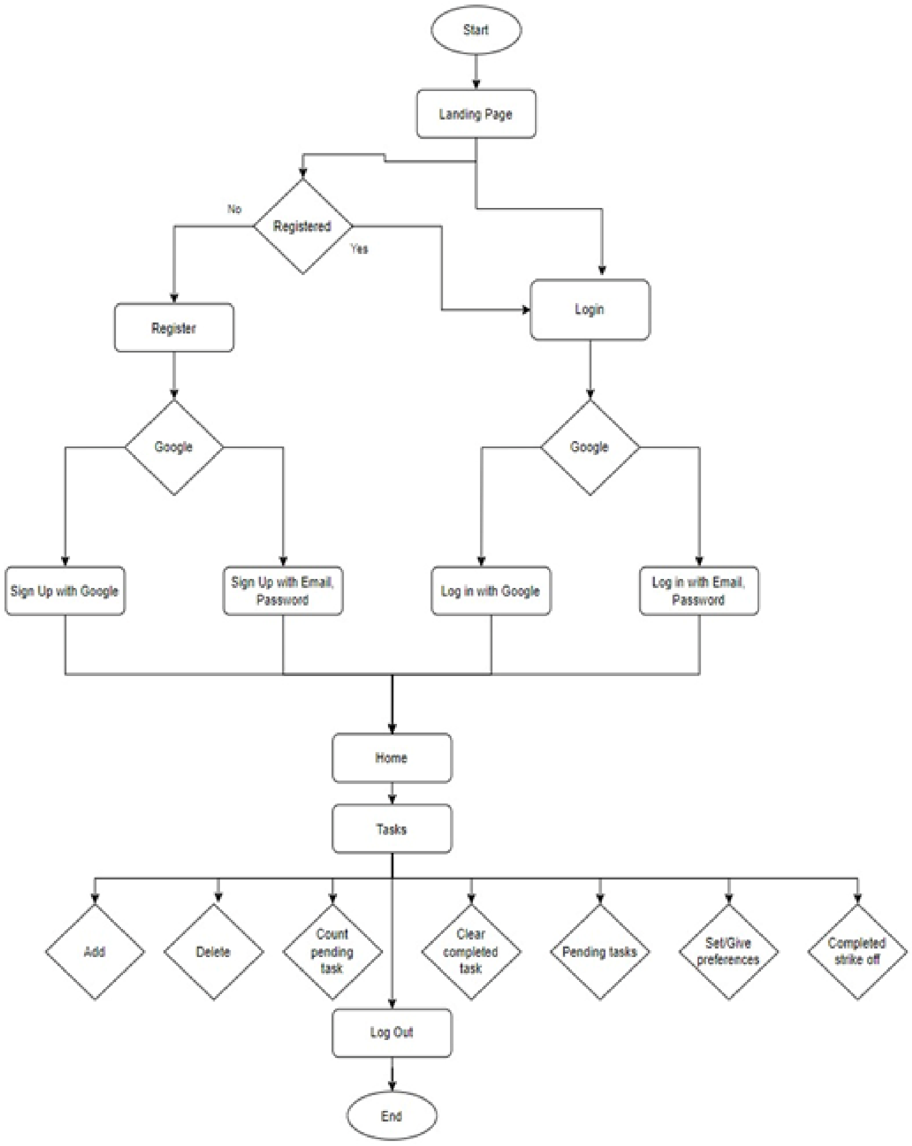
1. ***Unit Testing:***
   - *Write unit tests for components and functions.*
2. ***Integration Testing:***
   - *Test the integration of frontend, authentication, and backend components.*
3. ***End-to-End Testing:***
   - *Perform end-to-end testing to ensure the entire system works seamlessly*

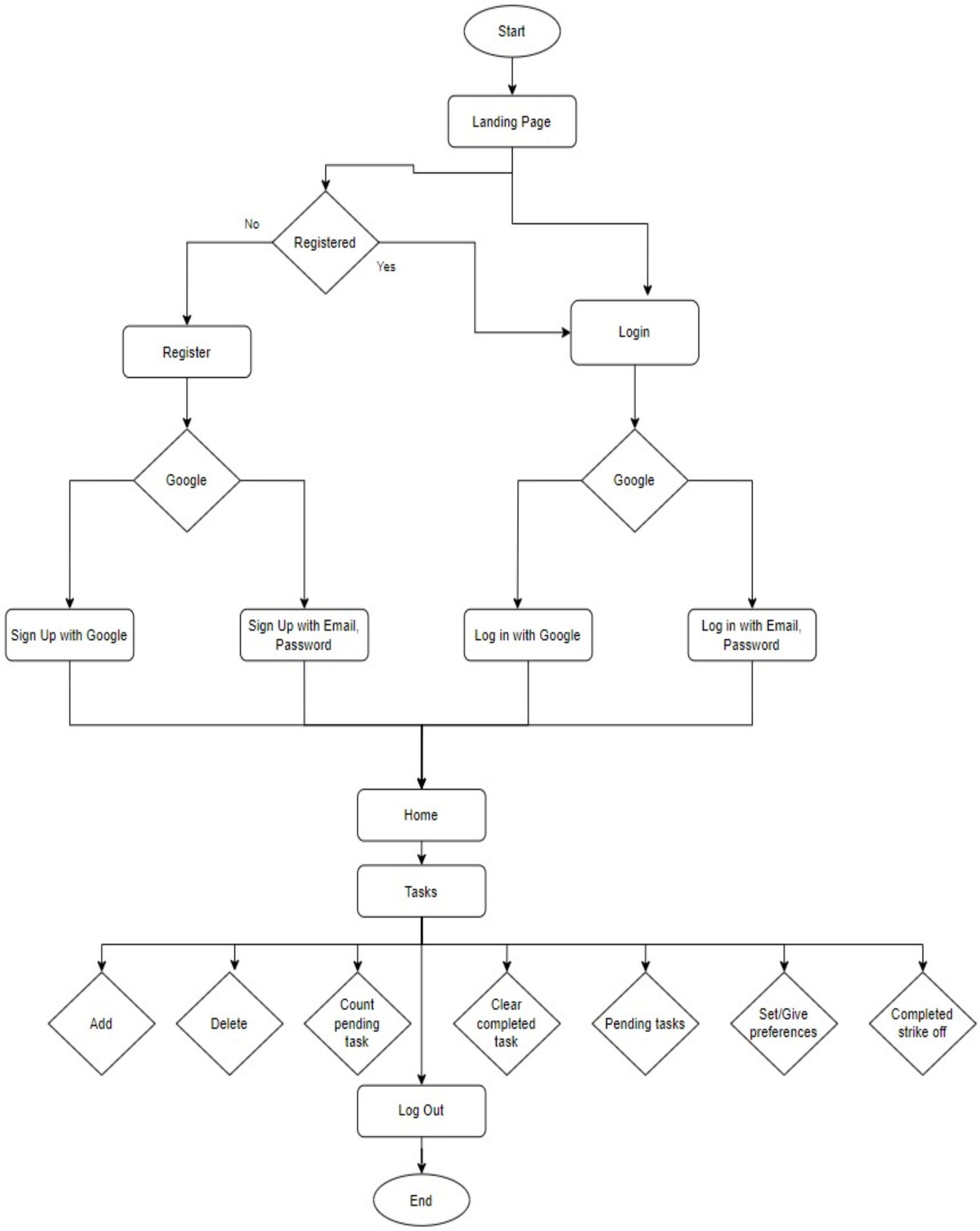
1.1 System Architecture

We have use REACT framework for frontend css for styling and FIREBASE for backend.


Landing Page
Login / Sign Up
Authenticate
User
Home
Allowing user to perform different operations on this task like add, delete, clearing, all, completed tasks, and settin preferences
Storing it into database accordingly
Saving Data
Log out

1.2 Module wise flow diagram





1.3 Software development tools

a. Application Development Tool:

- ***React.js:*** *A JavaScript library for building user interfaces. It is suitable for building the frontend of the Todo List application due to its component-based architecture and efficient state management.*

b. HTML Authoring Tool:

- ***Visual Studio Code:*** *A lightweight and powerful source code editor that supports HTML development with features like syntax highlighting, auto-completion, and integrated terminal.*

c. Word Processor for Documentation:

- ***Microsoft Word or Google Docs:*** *These tools can be used for documenting project requirements, design specifications, and user manuals. Collaborative editing in Google Docs may be beneficial for team collaboration.*

d. Tool for Drawing Diagrams:

- ***Draw.io:*** *A free, open-source tool for creating flowcharts, process diagrams, and other visual representations. Use it for designing system architecture, database schemas, and workflow diagrams.*

e. Automated Testing Tools:

- ***Jest and React Testing Library:*** *For testing React components and ensuring their functionality.*
- ***Firebase Emulator Suite:*** *To facilitate local testing of Firebase services, including Firestore.*

Additional Tools (Optional):

- ***Tailwind CSS IntelliSense Extension:*** *An extension for Visual Studio Code that provides autocompletion and linting for Tailwind CSS, improving the styling workflow.*

Post Lab Activities:

1. Design document is a very important in software development life cycle.

Comment on the above statement.

A design document is crucial in the software development life cycle as it serves as a blueprint, providing a comprehensive overview of the system's architecture, components, and functionalities. It aids in communication among team members, guides the implementation process, and facilitates future maintenance. The document acts as a reference point for developers, ensuring consistency and alignment with project goals, ultimately contributing to the efficiency and success of the software development process.