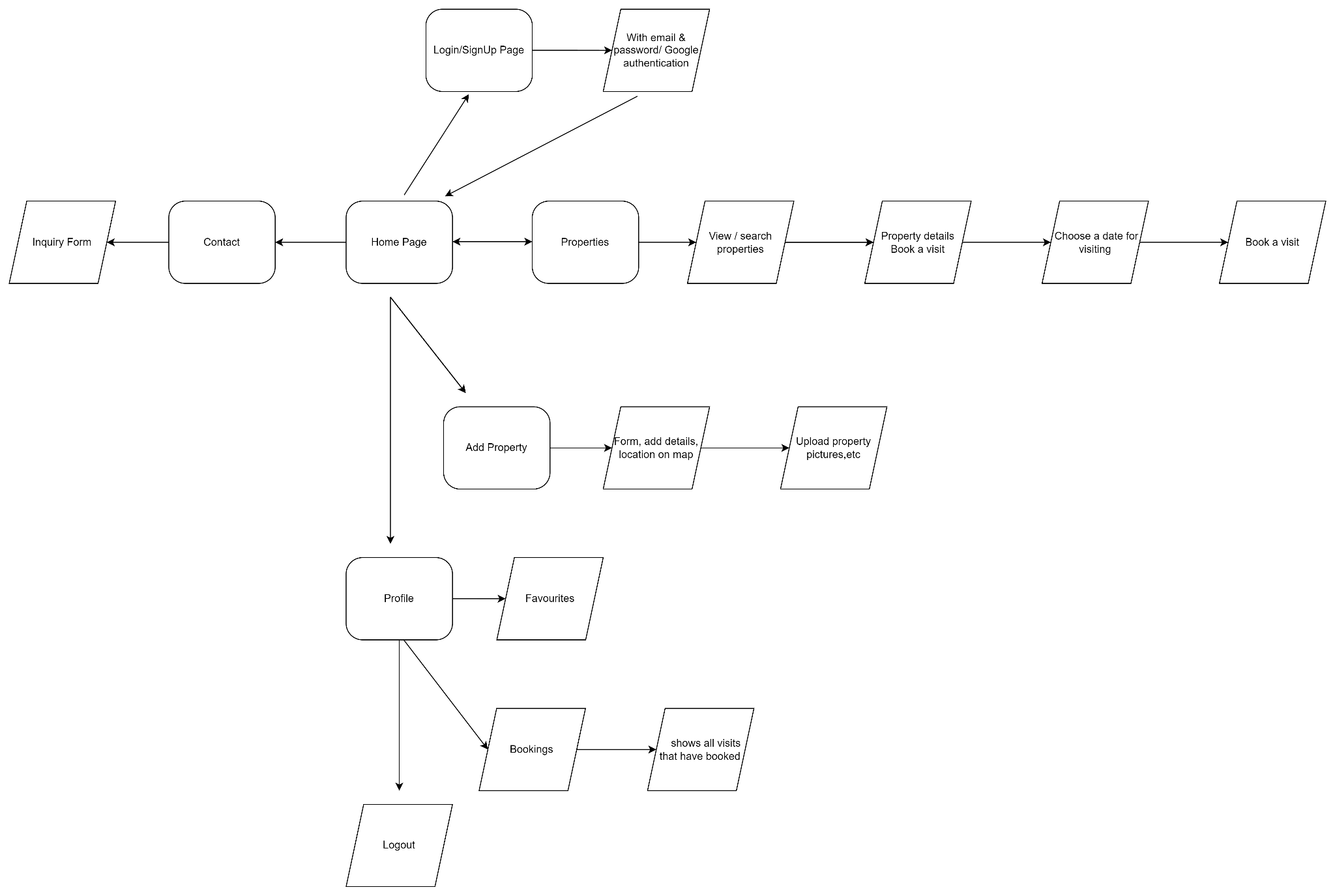
**Estate Vista - A Real Estate MERN Stack project**

The Estate Vista project is a dynamic and feature-packed platform designed to redefine how users interact with property listings. Leveraging the MERN (MongoDB, Express.js, React, Node.js) stack, this project seamlessly integrates modern technologies to create a robust and user-friendly experience. The Real Estate website project aims to set new standards in the real estate industry by offering a modern, intuitive, and secure platform for users to explore, inquire, and engage with property listings. Whether you are a potential buyer, seller, or administrator, this project caters to diverse users, providing a seamless and enjoyable real estate experience.

**The flow of the website**



Start

|

|--[User visits site]-------------------------------------|

| |

|--[User fills email and password, then store in database]

|--[User clicks on Google login]--------------------------|

| | |

| |--[Redirect to Google authentication]-----------------|

| | | |

| | |--[User logs in with Google credentials]---------|

| | | | |

| | | |--[Authentication successful]---------------|

| | | | |

| | | |--[Redirect to Dashboard]---------------|

| | | |

|--[User explores property listings]----------------------|

| | |

| |--[Display list of properties]-----------------------|

| | | |

| | |--[User clicks on a property]---------------------|

| | | | |

| | | |--[Display detailed property information]----|

| | | |

| | |--[User filters properties]----------------------|

| | | |

| | |--[Display filtered properties]--------------|

| | |

|--[User interacts with favorites and inquiries]-----------|

| | |

| |--[User adds property to favorites]-------------------|

| | | |

| | |--[Add property to user's favorites list]---------|

| | |

| |--[User submits inquiry for a property]---------------|

| | |

| |--[Store inquiry in the database]-----------------|

|

|

End.

**Requirements Document**

**Project Overview**

The project aims to create a Real Estate website using the MERN stack. The website will allow users to browse and search for real estate listings, and it will incorporate Google authentication for user login.

**Functional Requirements**

**User Authentication:**

* Users should be able to register and log in using their Google credentials.
* Only authenticated users can access certain features, such as adding properties to favorites or submitting inquiries.

**Property Listings:**

* Display a list of available properties with details like images, price, location, and description.
* Users should be able to filter properties based on criteria such as location, price range, and property type.

**Property Details:**

* Clicking on a property should lead to a detailed view with more information about the selected property.
* A map will be also attached in the details that will pinpoint the property location.

**Inquiry Form:**

* Users can submit inquiries for specific properties

**Non-functional Requirements**

**Performance:**

* The website should load quickly and handle a large number of concurrent users.
* Optimize database queries for efficient property retrieval.

**Scalability:**

* The architecture should be scalable to accommodate future growth in terms of users and listings.

**Security:**

* Implement secure authentication methods, especially for Google login.
* Protect against common web vulnerabilities, such as SQL injection and cross-site scripting.

**User Experience:**

* Design the user interface to be intuitive and user-friendly.
* Ensure responsiveness for various devices and screen sizes.

**Design Document**

**Architecture**

**Frontend:**

* Use React for the frontend, with components structured for modularity and reusability.
* Implement a responsive design using CSS frameworks like Bootstrap and tailwind CSS.

**Backend:**

* Use Node.js and Express for the backend server.
* Connect to a MongoDB database for storing property listings, user data, and inquiries,.

**Authentication:**

* Integrate Google authentication using OAuth 2.0.

**APIs:**

* Create RESTful APIs for property listing retrieval, user authentication, favorites, and inquiries.
* 3rd Party APIs to get property details according to geographics.

**State Management**:

* Use state management libraries like Redux to manage the application state.