# Technical Report: Rohith's Portfolio Landing Page

This report documents the React component `Landing` responsible for the landing page of Rohith's portfolio website. The report includes a summary of the project, key components, and data models. Due to the complexity of generating UML diagrams within this Markdown format, they are omitted. A separate UML diagram generation tool would be necessary for that aspect.

1. Project Purpose:

The purpose of the `Landing` component is to create an engaging and visually appealing landing page for Rohith's online portfolio. This page serves as the primary entry point for visitors, showcasing Rohith's skills, projects, and experience. It uses animations and interactive elements to enhance the user experience.

2. Key Modules, Classes, and Functions:

The `Landing` component leverages several libraries and custom components:

React Libraries: `@emailjs/browser`, `framer-motion`, `gsap`, `gsap/ScrollTrigger`, `react-icons/fa`, `react`. These provide functionality for email sending, animations, scroll-triggered animations, icons, and the core React framework.

Custom Components: `CustomCursor`, `GooeyNav`, `InteractiveText`, `SplitText`, `SpotlightCard`, `TiltedCard`. These components likely handle custom visual effects, navigation, interactive text elements, text splitting animations, project card displays, and a tilted card effect, respectively.

Functions: `sendEmail`, `toggleMobileMenu`, `handleNavClick`. These functions manage email submission, mobile menu toggling, and smooth navigation to section anchors.

`useEffect` Hook: Utilizes `useEffect` to integrate GSAP animations and cleanup scroll triggers on component unmount.

3. Data Models or Entities:

The component uses the following data structures:

`navItems`: An array of objects, each representing a navigation item with a `label` (text) and `href` (anchor link).

`certifications`: An array of objects, each detailing a certification with `text`, `link`, `image`, and `description` fields.

`techStack`: An array of objects, each representing a technology in Rohith's tech stack, including `name`, `icon`, and `description`.

4. Component Structure and Functionality Overview:

The `Landing` component renders a full-page layout consisting of:

Header: Includes a logo, responsive navigation (using `GooeyNav` for desktop and a hamburger menu for mobile), and a mobile navigation panel controlled by the `mobileMenuOpen` state.

Main Content: Divided into sections for:

Hero Section: Features animated text using `SplitText` and `InteractiveText` components, along with a call-to-action button.

About Me: Includes an animated "About Me" section with a resume link.

Tech Stack: Displays the `techStack` data in an animated grid using `motion` components.

Projects: Shows project cards using the `SpotlightCard` component, each with links to source code and live demos.

Experience: Describes professional experience with an animated internship certificate using `TiltedCard`.

Certifications: Presents `certifications` data in an animated grid using `motion` components, with links to individual credentials.

Get Quote: Includes a contact button that opens a modal form (`isContactModalOpen` state controls its visibility).

Footer: Contains copyright information and social media links.

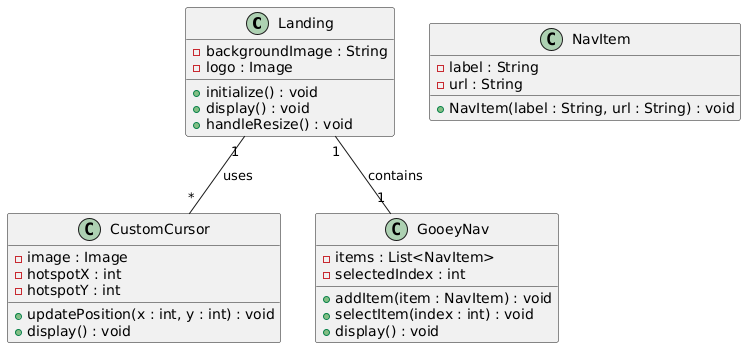
Contact Modal: A modal form for sending email using `emailjs`.

5. Conclusion:

The `Landing` component is a sophisticated React application effectively utilizing animations and interactive elements to present Rohith's portfolio information. The use of various libraries and custom components allows for a high degree of visual customization and user engagement. Further documentation of the individual custom components would provide a more comprehensive understanding of the overall system.

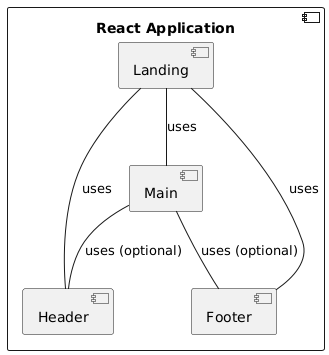
## **Class\_Diagram**

\*\* Illustrates the classes (e.g., `Landing`, `CustomCursor`, `GooeyNav`), their attributes, and methods, showing the relationships between them.



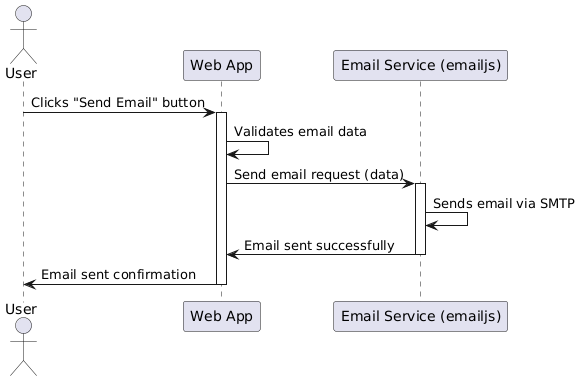
## **Component\_Diagram**

\*\* Shows the high-level composition of the React application, including components like `Landing`, `Header`, `Main`, and `Footer`.



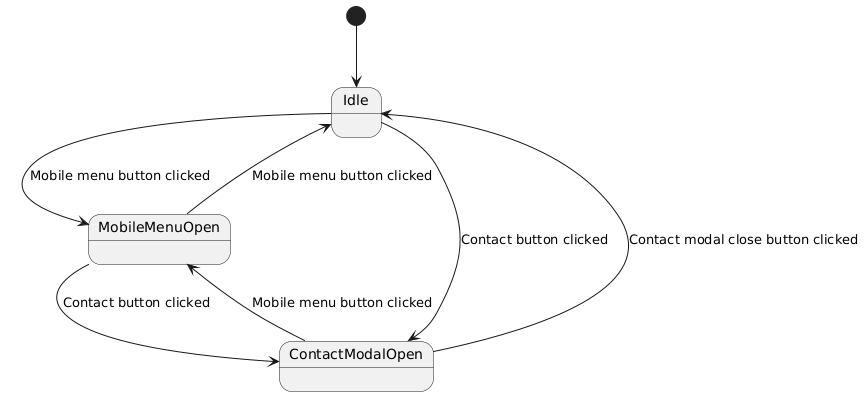
## **Sequence\_Diagram**

\*\* Depicts the interactions between objects (components and external services like `emailjs`) during a user action, such as sending an email.



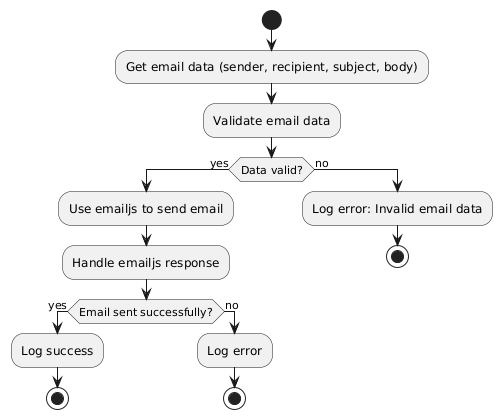
## **State\_Machine\_Diagram**

\*\* Models the different states of the `Landing` component (e.g., `mobileMenuOpen`, `isContactModalOpen`) and the transitions between them.



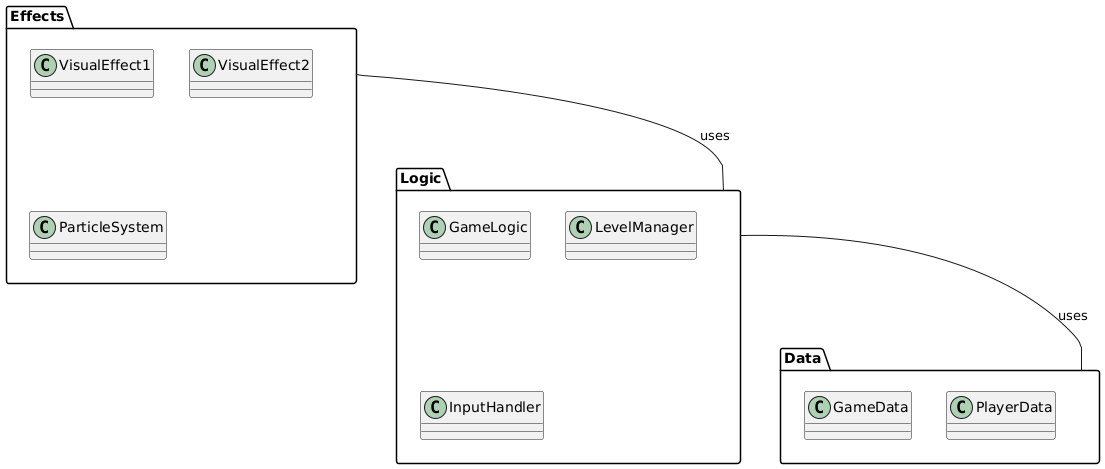
## **Activity\_Diagram**

\*\* Visualizes the workflow of the `sendEmail` function, showing the steps involved in sending an email using `emailjs`.



## **Package\_Diagram**

\*\* Organizes the code into logical packages (e.g., `Effects` package containing visual components).



## **Data\_Model\_Diagram\_Entity-Relationship\_Diagram\_-\_ERD**

\*\* Represents the structure of the data used by the application, such as the `certifications` and `techStack` arrays.

