**LinkForge**

**Design Document**

**Introduction**

The LinkForge project aims to create a web application that allows users to create a personalized landing page with multiple links to their social media profiles, websites, blog posts, and other online content. This project provides users with a convenient way to share multiple links through a single URL, making it easier for their audience to discover and navigate their online presence.

**Objectives**

* Provide users with a simple and intuitive interface for creating and customizing their link tree pages.
* Allow users to add, remove, and rearrange links on their pages.
* Implement authentication and user account management to allow users to save and edit their link tree pages.

**Key Features**

* User Registration and Authentication: Users can sign up for an account and log in to access their link tree dashboard.
* Link Management: Users can add, edit, and delete links on their link tree pages. They can also rearrange the order of links as needed.
* Responsive Design: The application is designed to be responsive and accessible on various devices, including desktops, tablets, and smartphones.
* Shareable URLs: Each user's link tree page has a unique URL that they can share with their audience on social media, websites, and other platforms.

**Technology Stack**

**Frontend:**

**React.js**: A JavaScript library for building user interfaces, providing a component-based architecture and efficient state management.

**React Router**: A routing library for React.js applications, facilitating navigation and routing between different components and views.

**Tailwind CSS:** A utility-first CSS framework for creating responsive and customizable user interfaces, enabling rapid development and easy styling.

**Backend:**

**Node.js**: A JavaScript runtime environment for building scalable and server-side applications, providing an event-driven architecture and non-blocking I/O operations.

**Express.js**: A minimalist web application framework for Node.js, used for creating robust and modular backend APIs and server-side logic.

**Database:**

**MongoDB:** A NoSQL database that stores data in flexible, JSON-like documents, providing scalability and flexibility for handling large volumes of data. MongoDB's document-oriented model is well-suited for storing user profiles, link data, and other structured information.

**Authentication:**

**JSON Web Tokens (JWT):** A standard for securely transmitting information between parties as a JSON object, used for user authentication and authorization. JWTs are commonly used in web applications to authenticate users and provide access to protected resources.

**Deployment**

**Render:** A cloud platform for hosting and deploying web applications, providing scalable infrastructure and continuous integration/deployment (CI/CD) capabilities for delivering the application to production environments.