**LinkForge**

**Design Document**

**1. 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.

**2. 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.

**3. 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.

**4. 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:**

The backend codebase can be deployed on platforms like Heroku or AWS, while the frontend can be hosted on Vercel or Netlify. Utilize Git for version control and collaboration, and deploy frontend changes seamlessly with Vercel's integration with GitHub repositories.

**5. Security**

Implement security measures to protect user data, such as encryption of passwords, input validation

**6. API Design**

The backend provides RESTful APIs for:

* **Links:**

1. POST /api/links: Adds a link
2. PUT /api/links/:id: Updates a link
3. DELETE /api/links/:id: Deletes a link

* **Login and Logout:**

1. POST /api/login: Logs in user
2. POST /api/logout: Logs out user

* **Signup and Refresh:**

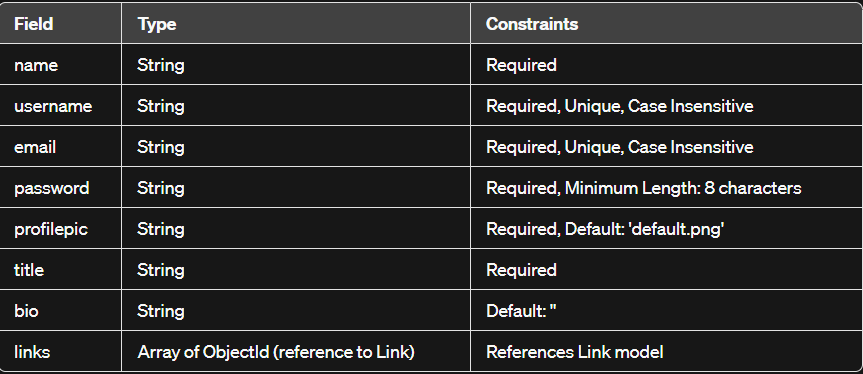
1. POST /api/signup: Create new user
2. POST /api/refresh: Refreshes access token

* **Users:**

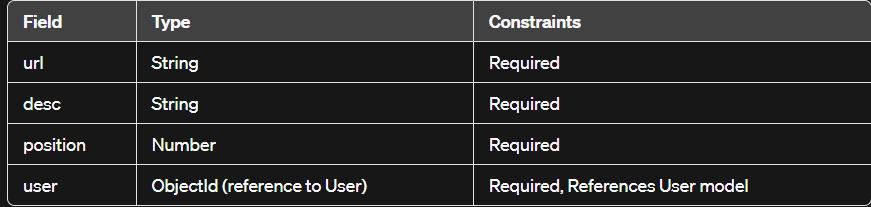
1. GET /api/users/:username: Gets a specific user
2. GET /api/users/:username/image: Gets profile picture of a specific user
3. PUT /api/users/:id: Updates specific user

**7. Database Design**

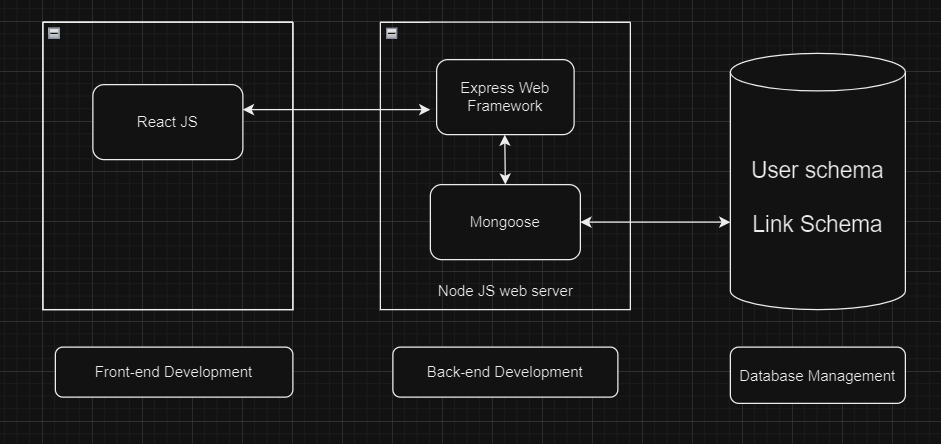
**User Schema**

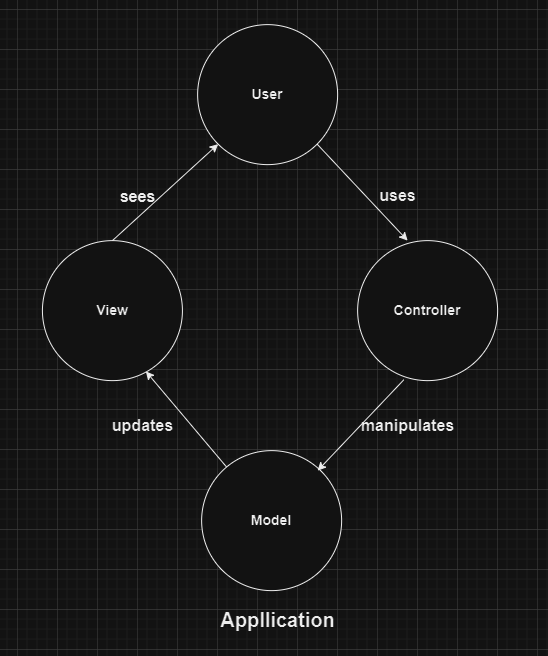


**Link Schema**

****

**8. Application Design**

****

****

**9. Maintenance and Support**

Regular maintenance and updates will be provided to address any bugs, security vulnerabilities, or performance issues.  
User feedback will be collected and incorporated into future updates to enhance the user experience and add new features as needed.

**10. Conclusion**

The LinkForge project aims to provide users with a convenient and customizable solution for creating link tree pages to showcase their online presence. By following a structured development approach and leveraging modern technologies, the project will deliver a high-quality web application that meets the needs of its users and provides a seamless experience across various devices and platforms.