**Project Overview**

This React project aims to provide developers with an easy solution for various components used on a website for example. Navbar, cards, carousel, footer etc. This project helps the developer to customize the given components with different customizations like texts and links without manually typing the code. They can simply customize the components and directly copy the code without much hassle with a single click. They can customize different components and position them according to themselves and copy the code for the whole page together. The application will leverage separate pages for different element customizations, a preview page, and a positioning page.

**Architecture**

***Frontend (React)***

- The frontend will feature distinct pages for element customizations.

- ***Navbar Page***: Choose from various Navbar styles and configurations.

- ***Cards Page***: Customize the appearance and content of cards.

- ***Carousel Page***: Configure carousel settings.

- ***Pricing Page***: Select and customize pricing components.

- ***Footer Page***: Customize the footer section.

- ***Preview Page***: View and copy the generated code/design.

- ***Positioning Page***: Arrange and position elements on the webpage.

***Backend***

- ***Authentication, Database, APIs***: Utilize JWT for authentication, MongoDB for storing comprehensive customization data, and expose APIs for CRUD operations on user-specific elements.

**Tech Stack**

***Frontend***

-React.js

-Tailwind CSS

***Backend***

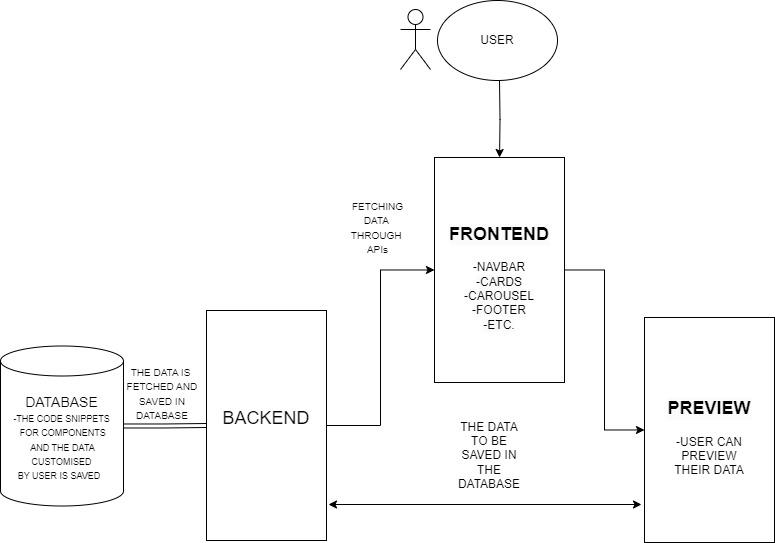
-Node.js

-MongoDB

-Express.js

***Authentication***

-JWT



**Components**

***Frontend Components***

1. **Navbar Page**

- Choose from various Navbar styles and configurations.

- Save selected Navbar options to the database.

2. **Cards Page**

- Customize the appearance and content of cards.

- Store card configurations in the database.

3. **Carousel Page**

- Configure carousel settings.

- Save carousel configurations to the database.

4. **Pricing Page**

- Select and customize pricing components.

- Store pricing configurations in the database.

5. **Footer Page**

- Customize the footer section.

- Save footer configurations to the database.

6. **Preview Page**

- Display a preview of the user's designed webpage.

- Retrieve stored configurations from the database for rendering.

7. **Positioning Page**

- Allow users to arrange and position elements on their webpage.

- Retrieve stored configurations from the database for positioning.

***Backend Components***

- **Authentication, Database Management, APIs**: Ensure that configurations for all elements are securely stored and retrieved from MongoDB.

***API Endpoints***

1. **Save Component Code**

- *Endpoint* : `POST /api/components`

- *Description* : Save the entire code for a component in the database.

- Request Body:

json

{

"name": "Main Navbar",

"type": "navbar",

"code": "<!-- Entire code for the navbar → ,

“Customizables” : {“customizable fields”},

}

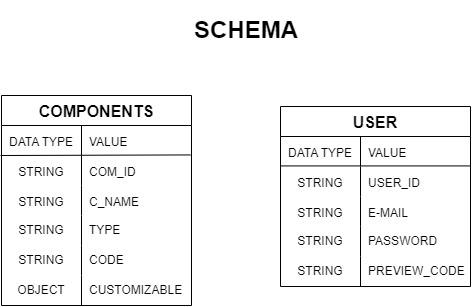
- Response: Returns the saved component document.

2. **Retrieve Component Code**

- Endpoint: `GET /api/components/:id`

- Description: Retrieve the entire code for a specific component and its customizable fields.

- Response: Returns the component document.



**Security Considerations**

- Implement secure data handling to protect user configurations.

- Ensure proper authorization for CRUD operations on user-specific data.

- Implement validation for user inputs on customization pages.

***Future Enhancements***

- Implement version control for designs in the database.

- Consider incorporating user access controls for collaborative editing.