# **Introduction**

## **Overview**

This project is an industry-level web application inspired by **Bit.ai**, enabling users to create proposals effortlessly within a **canvas-based editor**. The application offers a seamless document-building experience using simple **button clicks**, allowing users to add, modify, and structure content interactively.

In addition to document creation, the platform provides essential features like:

* **Workspaces**: Organize projects and collaborate with teams.
* **Templates**: Predefined structures to speed up document creation.
* **Sharing & Collaboration**: Secure sharing options for team members or external stakeholders.

## **Technology Stack**

The project is built using the **MERN** (MongoDB, Express.js, React.js, Node.js) stack, ensuring a **scalable and efficient**architecture.

Additionally, several key libraries enhance the functionality:

* **Slate.js**: A powerful editor framework for rich text editing.
* **React Router DOM**: Enables smooth navigation across the application.
* **Bcrypt**: Handles password hashing for secure authentication.
* **CodeMirror (or any coding text area library)**: Provides syntax-highlighted code editing.

This documentation will cover the application's architecture, features, implementation details, and best practices to ensure a clear understanding of its functionality.

# **Work Done**

## **Editor Module**

The **Editor Module** is the core of the application, allowing users to create and edit content interactively within a **canvas-based** environment. It consists of various fundamental elements that enhance the document creation experience.

### **Implemented Components**

Each element in the editor is designed as a separate **React component** for modularity and maintainability. The following components have been implemented:

* **Heading**: A component for adding section titles.
* **Paragraph**: Standard text content.
* **Double Paragraph**: A structured text block with two paragraphs.
* **Cost Module**: A specialized component for adding pricing details.
* **Image**: Allows users to insert standalone images.
* **Image with Paragraph**: Combines an image with descriptive text.

These components are built using **Slate.js** and stored in the **Slate folder**, ensuring a structured and maintainable editor module.

# **Pending Work**

While the core functionalities have been implemented, the following features are yet to be completed:

1. **Search Functionality**
   * Users should be able to search for specific content within the editor.
   * Implementation will involve **indexing text data** and integrating a **search bar**.
2. **Export & Download Options**
   * Users should be able to **download** their created proposals in **Word (DOCX) and PDF formats**.
   * Requires integration of libraries such as **jsPDF (for PDF)** and **mammoth.js (for Word conversion)**.
3. **Responsive Design Issues**
   * Currently facing UI issues on different screen sizes.
   * Need to improve responsiveness, possibly by using **CSS media queries** and **Flexbox/Grid layout optimizations**.