**Requirements Document**

* **Purpose:** Clearly outline the project goals and specific needs for a rice mill management system.
* **Contents:**
  + **Project Overview:** Describe the project's purpose and scope.
  + **Functional Requirements:** Detailed list of features, like inventory tracking, order management, production tracking, etc.
  + **Non-Functional Requirements:** Performance expectations, security needs, scalability, etc.
* **Benefits:** Provides a clear vision and a reference to ensure you’re on track during development.

**2. Technical Specification Document**

* **Purpose:** Detail the technical architecture and chosen technologies.
* **Contents:**
  + **Architecture Overview:** Explain the overall structure (frontend, backend, database).
  + **Database Schema:** Show tables or collections, with relationships between them.
  + **API Documentation:** Describe each API endpoint, including parameters, request methods, and response formats.
  + **Technology Stack:** List chosen frameworks, libraries, and tools with reasons for their selection.
* **Benefits:** Helps current and future developers understand the system’s structure and integrate new features.

**3. User Interface (UI) Documentation**

* **Purpose:** Provide design guidelines and detailed descriptions of each interface.
* **Contents:**
  + **Wireframes or Mockups:** Show page layouts and component designs.
  + **User Flow Diagrams:** Illustrate how users navigate through the system.
  + **Component Breakdown:** Describe the purpose of each component and interaction.
* **Benefits:** Ensures a consistent user experience and can act as a guide when implementing the UI.

**1. Project Overview Document**

* **Purpose**: Explain the purpose and goals of your rice mill management system, including the problems it aims to solve.
* **Audience**: General (stakeholders, developers, potential users).
* **Contents**:
  + Project description and objectives.
  + Key features and benefits.
  + Overview of the MERN stack and why it was chosen.

**2. Requirements Specification Document**

* **Purpose**: Outline all functional and non-functional requirements of the system.
* **Audience**: Developers and stakeholders.
* **Contents**:
  + **Functional Requirements**: Specific functionalities your system should have (e.g., inventory management, production tracking, user roles, and order handling).
  + **Non-Functional Requirements**: Performance requirements, scalability, security needs, and reliability.

**3. Database Design Document**

* **Purpose**: Describe the database structure and relationships between entities.
* **Audience**: Backend developers and database administrators.
* **Contents**:
  + Overview of MongoDB collections.
  + Detailed schema definitions for each collection (e.g., Inventory, Orders, Production).
  + Explanation of relationships between collections, if any.
  + Indexing strategy, if applicable.

**4. API Documentation**

* **Purpose**: Provide detailed information on each API endpoint, enabling developers to integrate or use your backend services.
* **Audience**: Frontend developers, third-party integrators.
* **Contents**:
  + Description of each endpoint (method, URL).
  + Request parameters (body, headers, URL params).
  + Sample requests and responses.
  + Error codes and possible responses.
  + Authentication details (e.g., JWT requirements).

Tools like **Swagger** or **Postman** can help you generate interactive API documentation automatically.