**Task Manager Documentation**

**1. Introduction**

* **Purpose**: To provide an overview of the Task Manager application.
* **Scope**: This document outlines the features, user roles, and functionality of the Task Manager.

**2. User Roles**

* **Admin/Manager**:
  + Can create, edit, assign, and delete tasks.
  + Can view the status of all tasks.
  + Can manage user accounts (create, update, delete).
* **Employee**:
  + Can view tasks assigned to them.
  + Can update task status (To Do, In Progress, Completed).

**3. Functional Requirements**

**3.1 User Authentication**

* **Signup**:
  + Users can create an account by providing necessary details (name, email, password, etc.).
  + Email verification for new users.
* **Login**:
  + Users can log in using their registered email and password.
  + Password recovery option via email.

**3.2 Task Management**

* **Create Task**:
  + Admin/Manager can create new tasks with details like title, description, due date, priority, and assignee.
* **Edit Task**:
  + Admin/Manager can edit task details.
* **Delete Task**:
  + Admin/Manager can delete any task.
* **Assign Task**:
  + Admin/Manager can assign tasks to employees.
  + Notifications sent to employees upon task assignment.

**3.3 Task Status Management**

* **Mark Task Status**:
  + Employees can update the status of tasks (To Do, In Progress, Completed).
* **Task Filters**:
  + Ability to filter tasks based on status, priority, due date, etc.

**4. Non-Functional Requirements**

* **Security**:
  + User data and task details must be securely stored and transmitted.
  + Implement role-based access control.
* **Performance**:
  + The application should be responsive and handle concurrent users efficiently.
* **Usability**:
  + The user interface should be intuitive and easy to navigate.

**5. User Interface**

* **Login/Signup Page**:
  + Forms for user registration and login.
* **Dashboard**:
  + Overview of tasks and their statuses.
  + Widgets for task summary and notifications.
* **Task Management**:
  + Forms for creating and editing tasks.
  + Task list with filtering and sorting options.

**6. Database Design**

* **User Table**:
  + Fields: user\_id, name, email, password, role, created\_at, updated\_at.
* **Task Table**:
  + Fields: task\_id, title, description, due\_date, priority, status, assigned\_to, created\_by, created\_at, updated\_at.

**7. API Endpoints**

* **User Authentication**:
  + POST /signup
  + POST /login
  + POST /logout
* **Task Management**:
  + POST /tasks
  + GET /tasks
  + PUT /tasks/{id}
  + DELETE /tasks/{id}
* **Task Status Update**:
  + PATCH /tasks/{id}/status

**8. Implementation Details**

* **Frontend**:
  + Framework: (e.g., Vue.js, React)
  + Responsive design principles.
* **Backend**:
  + Framework: Laravel
  + RESTful API development.
* **Database**:
  + MySQL/PostgreSQL
  + Eloquent ORM for database interactions.

**9. Deployment**

* **Hosting**: (e.g., AWS, Heroku)
* **Continuous Integration/Deployment (CI/CD)**: (e.g., GitHub Actions, Jenkins)
* **Monitoring and Logging**: (e.g., New Relic, Loggly)