**Task Management Application Report**

**Introduction**

The Task Management Application is designed to help users manage their tasks effectively. It provides features such as creating, updating, and deleting tasks, along with a responsive user interface for seamless usage across devices.

**Technical Stack**

* Backend: Node.js, Express.js, MongoDB, Mongoose
* Frontend: React.js, Axios for API requests
* Additional Tools: npm for package management, dotenv for environment variables

**Installation Instructions**

1. Clone the repository from GitHub.
2. Navigate to the project directory in your terminal.
3. Run npm install to install dependencies.
4. Set up MongoDB Atlas and obtain the connection string.
5. Create a .env file in the root directory and add the MongoDB connection string.
6. Run npm start to start the server and the React frontend.

**Code Overview**

**Server.js**

* Sets up the Express server, middleware, and routes.
* Handles MongoDB connection using Mongoose.
* Includes error handling middleware.

**Task Model (task.js)**

* Defines the Mongoose schema for tasks, including title, description, due date, and completion status.

**Task Routes (tasks.js)**

* Defines CRUD operations for tasks using Express Router.

**Frontend Components**

* TaskForm.js: Handles form input for creating and editing tasks.
* TaskList.js: Displays a list of tasks fetched from the backend.
* TaskDetail.js: Shows detailed information of a single task.

**Challenges Faced**

* MongoDB Connection Issues: Resolved by updating the Mongoose connection options and using the correct MongoDB URI.
* Frontend Integration: Ensured seamless communication between frontend and backend using Axios for API requests.

**Future Improvements**

* User Authentication: Implement user authentication and authorization for task management.
* Task Categories: Add support for categorizing tasks and filtering based on categories.
* Notifications: Integrate notification system for task reminders and updates.