**VISHAL NISHAD (MCA- Graphic era hill)**

This is the link for the github repository Front end + Backend

<https://github.com/Xemb0/Task2Do>

This is link for the live demo WEBSITE TASK2Do

<https://task2do-vishalnishad.netlify.app/>

**Task2Do Project: Challenges, Design Choices, and Future Improvements**

**Introduction**

The Task2Do project is a web-based task management application developed using Vite + React on the frontend and Node.js + Express + MongoDB on the backend. This document discusses the challenges faced during development, the design choices made, and recommendations for future improvements.

**Challenges Faced**

1. **Backend Challenges: Separating Tasks by Weekday or Month**
   * **Issue**: One of the main challenges was implementing a feature to categorize tasks by weekday or month.
   * **Solution**: We developed a custom algorithm to categorize and display tasks based on the current week or month.
   * **Lesson Learned**: Efficient data querying and algorithm design are crucial for sorting and categorizing tasks effectively.
2. **Responsiveness**
   * **Issue**: Ensuring that the application is responsive across different devices and screen sizes was a significant challenge.
   * **Solution**: We used CSS media queries and followed responsive design principles to ensure the layout adapts well to various screen sizes.
   * **Lesson Learned**: Continuous testing and improvements are essential to achieve consistent responsiveness across devices.

**Design Choices**

1. **Frontend Architecture**
   * **Technology Stack**: We chose React.js with Vite for its fast development and support for modern JavaScript features.
   * **Styling**: CSS Modules were used for scoped styling, and we used Figma for designing and prototyping.
   * **State Management**: We employed the React Context API for managing the global state of the application.
2. **Backend Architecture**
   * **Technology Stack**: Node.js with Express.js and MongoDB were selected to ensure scalable and efficient data storage.
   * **Database**: MongoDB Atlas was used for cloud-based database hosting and management.

**Future Improvements**

1. **Enhanced Task Management Features**
   * **Implement**:
   * **Optimizing**: