Task Management App Development Report

Table of Contents

[Introduction 1](#_Toc180279067)

[Features 1](#_Toc180279068)

[Development Process 2](#_Toc180279069)

[1. Setting Up the Environment 2](#_Toc180279070)

[2. Building Core Components 2](#_Toc180279071)

[3. Implementing Sorting Logic 2](#_Toc180279072)

[4. UI and Styling Enhancements 2](#_Toc180279073)

[Conclusion 3](#_Toc180279074)

# Introduction

This report provides a detailed summary of the development process of a Task Management Application. The application was built using React and Next.js, with the core functionality centered around task management, including task creation, editing, deletion, and priority-based sorting. The app was designed with responsive layouts using Flexbox, and it incorporates a 3D card design for tasks, alongside animations and smooth transitions to enhance user interaction.

# Features

The application includes the following key features:

1. Task Creation: Users can create tasks by entering a title, description, and priority (high, medium, or low). The task is then added to the task list.

2. Task Editing: Each task is editable, allowing users to modify the title, description, and priority of any task.

3. Task Deletion: Tasks can be easily removed from the list using the delete functionality.

4. Mark Task as Completed: Users can toggle between marking tasks as completed or pending.

5. Priority-Based Sorting: The task list is automatically sorted by priority, ensuring that high-priority tasks appear at the top, followed by medium and low-priority tasks.

6. Responsive Layout: The app ensures two tasks are displayed per row, with a responsive design that adjusts the layout based on screen size.

7. 3D Card Design and Animations: Task items are displayed using a 3D card design, with hover effects and smooth transitions for an enhanced user experience.

# Development Process

## 1. Setting Up the Environment

The project was initiated with Next.js, chosen for its server-side rendering (SSR) capabilities and flexibility in building modern web applications. React was used to manage the components, and TypeScript was utilized to ensure type safety throughout the development process.

## 2. Building Core Components

The app is divided into two main components: `TaskForm` for creating tasks and `TaskList` for displaying them.

- TaskForm Component: This component allows users to input task details, including the title, description, and priority. Once a task is submitted, the form resets, and the new task is added to the task list.

- TaskList and TaskItem Components: The task list is built using Flexbox to display tasks in a two-column layout. Each task is presented as a 3D card, and the layout is fully responsive, ensuring optimal display on various devices.

## 3. Implementing Sorting Logic

One of the essential requirements of the app was to automatically sort tasks based on their priority. The tasks are sorted in real-time using JavaScript’s `sort` method, ensuring that tasks with higher priority appear at the top. This sorting logic is applied every time a task is added, edited, or deleted.

## 4. UI and Styling Enhancements

The UI was designed to be minimalistic but visually appealing. Using CSS, the task items were styled as 3D cards with shadows and rounded corners. Animations were added to create a hover effect, where the task cards appear to lift off the page slightly when hovered over, giving the app a polished and interactive feel.

A gradient background was applied to the buttons to match the color scheme of `#cdb4db`, `#ffc8dd`, `#ffafcc`, `#bde0fe`, and `#a2d2ff`. Button text was set to black for better contrast, and transitions were added to create smooth visual feedback when interacting with elements.

# Conclusion

The development of the Task Management Application successfully met all specified requirements. The app is fully functional with a clean, responsive design and an intuitive interface. Its flexibility allows for easy scalability and further enhancements, such as adding more advanced features like task deadlines, reminders, or category filtering. The combination of Next.js, React, and modern CSS techniques ensures that the application performs efficiently across devices.





