Task Manager Web App

A minor project report

Submitted by

Ashvini Konade

In the partial fullfillment of the degree BCA in the

Chandigah University

Acknowledgement

I would like to express my sincere gratitude to Mr. Shivkumar Konade for his invaluable guidance, unwavering support, and expert advice throughout the development of the Task Manager Web App. His insightful suggestions and encouragement have been instrumental in shaping this project.

I am also immensely thankful to Chandigarh University for providing me with the opportunity to undertake this project as a minor project. The academic environment and resources at Chandigarh University have played a pivotal role in nurturing my skills and providing the platform to explore and implement real-world solutions.

This project would not have been possible without the continuous support and encouragement from both Mr. Shivkumar Konade and Chandigarh University. Their contributions have been invaluable in the successful completion of this endeavor.

Abbreviations

HTML	:	HyperText Markup Language
CSS	:	Cascading Style Sheets
BCA	:	Bachelor of Computer Applications
PC	:	Personal Computer
IDE	:	Integrated Development Environment
APIs	:	Application Programming Interfaces
CRUD	:	Create, Read, Update, Delete

References

MDN Web Docs (https://developer.mozilla.org/): Provides comprehensive documentation on HTML, CSS, JavaScript, and web development concepts.
W3Schools (https://www.w3schools.com/): Offers tutorials, references, and exercises on web development technologies.
Stack Overflow (https://stackoverflow.com/): A community-driven platform for asking questions and finding solutions related to programming, including JavaScript and web development. **GitHub** (https://github.com/): Source for open-source projects and repositories, offering code references and examples.
CSS-Tricks (https://css-tricks.com/): Offers articles, guides, and tutorials on CSS, including modern layout techniques and design ideas.
Node.js Official Documentation (https://nodejs.org/en/docs/): Provides comprehensive documentation and resources for Node.js.
Vercel Documentation (https://vercel.com/docs): Offers guides and documentation for deploying web projects using Vercel.

Guide

Shivkumar Konade: A flutter mobile app developer (3+ years of experience). Chinchpur, South Solapur, Solapur.
BCA. +919284103047 shivkumarkonade@gmail.com
Linkedin : https://www.linkedin.com/in/shivkumar-konade-a39837196/

CERTIFICATE

This is to certify that this project entitled “**Task Manager Web App**” submitted in partial fullfillment of the degree of **BCA** to the **Chandigarh University** done by Ms Ashvini Konade Student ID: O21BCA16268 is an authentic work carried out by her under my guidance. The matter embodied is this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

A.A. Konade [illegible]

Signature of the student Signature of the guide

Date: 15/12/2023

Synopsis

The Task Manager Web App project aimed to develop a user-friendly web application using HTML, CSS, JavaScript, and Node.js for efficient task management. The project successfully created a platform enabling task creation, update, and deletion. While meeting its core objectives, limitations such as time constraints and technical expertise influenced the project's scope. Future considerations involve expanding functionalities and refining the application for improved task organization and productivity.

Objective and Scope

Objective:

The primary objective of the Task Manager Web App project is to develop a user-friendly and efficient web-based application aimed at enhancing productivity through effective task management. The project aims to leverage HTML, CSS, JavaScript, and Node.js technologies to create a platform that allows users to create, update, and delete tasks, thereby providing a streamlined and organized approach to managing tasks.

Scope:

The scope of the Task Manager Web App encompasses the following key features:
Task Creation: Users can create new tasks by providing task names, descriptions, due dates, and status.
Task Update and Deletion: Users can update existing tasks by modifying task details or delete tasks when no longer required.
User Interface: An intuitive and responsive user interface facilitating easy task management and interaction.
Backend Data Storage: Utilization of Node.js for backend functionality to store and manage task data.
Basic Task Operations (CRUD): Implementation of fundamental CRUD (Create, Read, Update, Delete) operations for tasks.

The project aims to deliver a functional and practical task management system, providing users with the tools necessary for effective organization and productivity enhancement.

Problem Definition

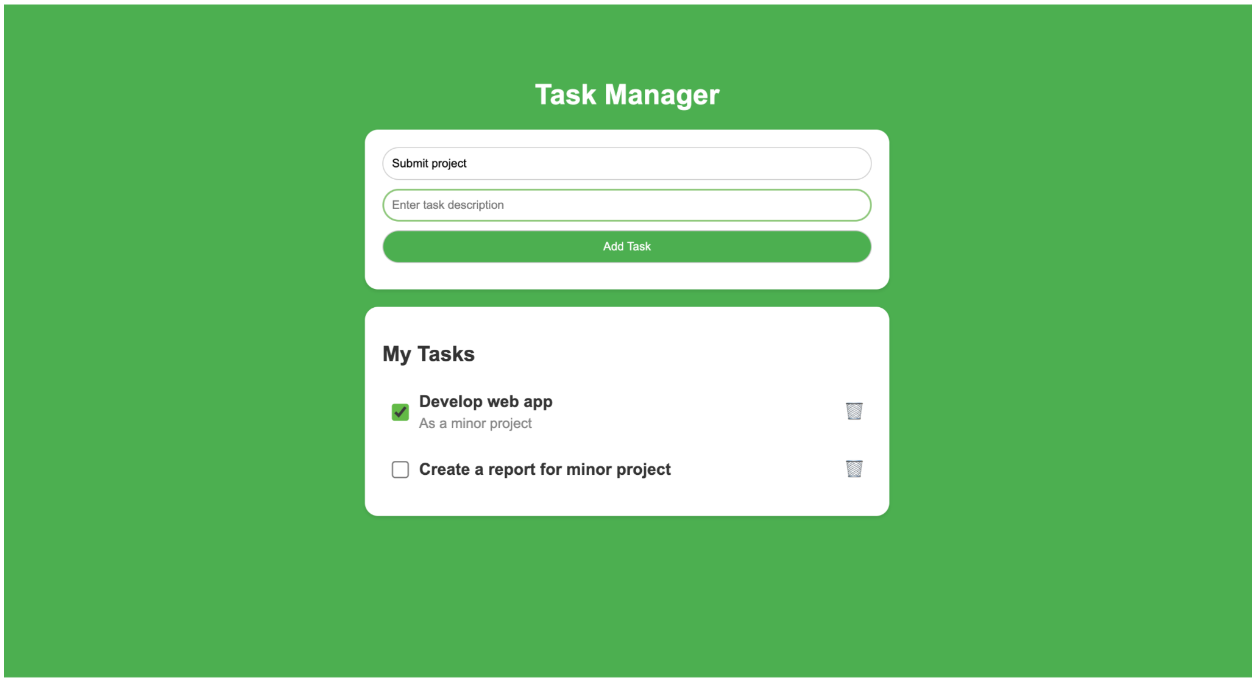
In today's fast-paced lifestyle, individuals often encounter difficulties in:

Task Organization: Managing multiple tasks and activities without a centralized system leads to disorganization and potential task oversight.
Task Prioritization: Lack of a structured approach to prioritize tasks often results in confusion about what needs immediate attention and what can be postponed.
Information Overload: With numerous tasks and information scattered across different platforms, individuals struggle to consolidate and manage their task-related data effectively.
Productivity Impact: Inefficient task management affects overall productivity and time management, leading to missed deadlines or incomplete tasks.

Task Manager Web App aims to offer a solution that streamlines task management, thereby enhancing organization, productivity, and time management for users.

Process Description

UI design: Chose a refreshing green color for the background to evoke a sense of productivity, growth, and positivity. Utilized CSS to design task items resembling white cards, providing a clear visual separation and a structured layout for each task. Rounded the corners of the cards for a softer and visually appealing look. Styled the input fields for task name and description with rounded edges, subtle borders, and padding for a cohesive look and feel. Applied CSS to create a button with a contrasting color, rounded corners, and padding for visual distinction and ease of interaction. Formatted checkboxes within the task cards to denote task completion. Styled them to be easily clickable and visually noticeable. Incorporated delete icons at the end of each task card, designed using CSS or Material Icons, for users to remove specific tasks conveniently. Maintained consistent font styles and sizes for better readability across the interface. Ensured proper alignment and spacing between elements to provide a visually organized and easy-to-navigate layout. Ensured the design elements were responsive, adjusting seamlessly to various screen sizes, thus ensuring a consistent and user-friendly experience across devices.



My Tasks

- [x] **Develop web app**
  As a minor project
- [ ] **Create a report for minor project**

Schema:

Task Schema: Describes the structure of each task entity. In this project, it might include fields like **id** (unique identifier), **name** (task name), **description** (task details), and **completed** (status of task completion).

Data Organization: The schema defines how tasks are stored and organized, either in memory (temporary storage) or a database (permanent storage). For instance, an array of objects might represent tasks, where each object follows the task schema.

Structure

Frontend Structure: This pertains to the organization of the user interface elements. For instance, the project uses HTML for structuring the layout, CSS for styling, and JavaScript for functionality. The structure ensures a clear and intuitive user experience, with input fields, buttons, and a display area for tasks.

Backend Structure: In a simple backend using Node.js, there might be routes or endpoints (APIs) for handling CRUD operations (Create, Read, Update, Delete) on tasks. Each endpoint would correspond to a specific task operation.

APIs (Application Programming Interfaces):

CRUD APIs: These APIs define the endpoints or routes that handle different task operations. For instance:

1. GET /tasks: Retrieves all tasks.
2. POST /tasks: Creates a new task.
3. PUT /tasks/:id: Updates an existing task.
4. DELETE /tasks/:id: Deletes a task by its unique identifier.

Data Communication: APIs serve as a communication bridge between the frontend and backend. The frontend interacts with these APIs to send requests (e.g., adding a task) and receive responses (e.g., fetching tasks).

Request-Response Format: These APIs typically follow RESTful principles, specifying how clients (frontend) can interact with the server (backend) using standard HTTP methods (GET, POST, PUT, DELETE) and data formats (JSON, XML).

Manual Testing: I spent time trying out the app myself to see if everything worked right. I found lots of things that weren't working like they should, so I fixed them. It took a while, but now the app works much better.

Deployment: App has been deployed successfully on Vercel platform which is hosting platform for frontend and backend projects.

Backend endpoint: https://taskmanager-jet-one.vercel.app/tasks

Frontend app url: https://taskmanagerapp-seven.vercel.app

Git repo: https://github.com/ASHVINI702090/task-manager-web-app

Git repo deployed branch (contain endpoint changes):
https://github.com/ASHVINI702090/task-manager-web-app/tree/deploy

Code can be run locally also.
To run backend:

```
cd backend
node server.js
```

After that open index.html from frontend folder.

Resources:

Project Team: Limited to a single developer with proficiency in HTML, CSS, JavaScript, and basic knowledge of Node.js.

External Guidance: Support and guidance from mentor Shivkumar Konade.

Development Tools: Utilizing Visual Studio Code as the primary Integrated Development Environment (IDE).

Online Documentation and Tutorials: Access to online resources like MDN Web Docs, W3Schools, and GitHub Guides for reference and learning.

Infrastructure: Local Development Environment: Utilizing a Windows-based PC for development and testing purposes. Web Server for Deployment: Planning to host the application on a web server for accessibility.

Limitations:

Time Constraint: Limited timeframe due to project submission deadlines may restrict the inclusion of advanced features or extensive testing.

Limited Backend Proficiency: Moderate knowledge in Node.js might limit the complexity of backend functionalities.

Scope Limitation: Project scope constrained to basic CRUD operations; advanced features might not be feasible within the given timeframe and expertise.

Single Developer: Limited human resources may impact the pace and scope of development.

Hardware Limitations: Limited access to high-performance hardware might affect the application's performance during testing.

Conclusion

The development of the Task Manager Web App project has been a valuable learning experience, combining theoretical knowledge with practical application in the realm of web development. Throughout this project, I have gained a deeper understanding of HTML, CSS, JavaScript, and Node.js, honing my skills in frontend and backend development.

The application successfully achieves its primary objective of providing users with a platform for efficient task management. With functionalities encompassing task creation, updating, and deletion, the Task Manager Web App offers a simple yet effective solution to organize and prioritize tasks.

While the project successfully met its core objectives, there remains ample scope for further enhancements and refinements. Future iterations could focus on expanding the application's functionalities to include task prioritization, user authentication, and task categorization. Additionally, addressing the identified limitations, such as backend proficiency and time constraints, would contribute to a more robust and feature-rich application.

In conclusion, the Task Manager Web App project stands as a testament to the application of acquired knowledge, highlighting the potential for continued growth and development in the field of web-based application development.