TO DO LIST USING NODE.JS

A Micro Project Report

Submitted by

R.NITHISH SELVAM

Reg.no: 99220040665

B.Tech - COMPUTER SCIENCE AND ENGINEERING,
Specialization- Artificial intelligence and Machine learning

Kalasalingam Academy of Research and Education

(Deemed to be University)

Anand Nagar, Krishnankoil - 626 126

FEBRUARY 2024

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Bonafide record of the work done by Nithish Selvam – 99220040665 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Specialization of the Computer Science and Engineering, during the Academic Year – Even Semester (2023-24)

Dr.P.PandiSelvam

Project Guide

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING

Kalasalingam Academy of Research and Education

Krishanan Kovil- 626126

Mr.Gnana Kumar

Faculty In-Charge

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING

Kalasalingam Academy of Research and Education
Krishanan Kovil- 626126

Mrs.K.Deepa Lakshmi

Evaluator

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING

Kalasalingam Academy of

Research and Education

Krishnan kovil - 626126

Abstract

Efficient task management solutions are more important than ever in the fast-paced world of today. In order to give customers a simple and efficient platform for organizing their daily chores, this project demonstrates the development of a To-Do List application using Node.js, a well-liked JavaScript runtime. The goal is to create a scalable, RESTful API that enables CRUD (Create, Read, Update, Delete) operations for task management by utilizing the asynchronous, event-driven architecture of Node.js and Express.js for server-side routing. The non-blocking nature of Node.js and the ease of use of JavaScript on both the client and server sides improve the application's performance, making it a good option for developers who want to create scalable and effective web applications. A good option for developers wishing to create scalable and effective web applications, Node.js's non-blocking architecture and JavaScript's ease of use on the client and server sides improve the application's performance.

Contents

Chapter 1

Introduction

1.1 Why do we need to do list?

We need To-Do Lists as a fundamental tool to navigate our busy lives with clarity and purpose. They bridge the gap between our goals and daily actions, serving as a roadmap for what we intend to accomplish. By laying out tasks in a structured format, To-Do Lists help in prioritizing what's important, ensuring that we focus our efforts on tasks that align with our objectives.

This organization aids in mitigating the overwhelm that comes from juggling multiple responsibilities, thereby reducing stress and enhancing our mental well-being. Moreover, the act of checking off completed tasks fosters a sense of progress and achievement, fueling motivation and driving us forward. In essence, To-Do Lists are not just about managing tasks; they're about cultivating a disciplined approach to productivity and personal growth.
This boosts motivation and helps us navigate our daily lives with more focus and less stress, making To-Do Lists an indispensable tool for anyone looking to improve their organizational skills and overall efficiency.

1.2 About the project

The To-Do List project is a web application designed to help users efficiently manage their tasks and enhance productivity. Utilizing Node.js for the backend, with Express.js for routing, the project embodies a full-stack JavaScript solution. Users can interact with a simple yet intuitive interface to add, view, edit, and delete tasks, essentially supporting the full spectrum of CRUD (Create, Read, Update, Delete) operations. The application focuses on providing a seamless user experience, enabling task prioritization and organization through a user-friendly web interface. By leveraging modern web technologies, this project not only serves as a practical tool for end-users but also as a learning platform for developers to understand the fundamentals of building and deploying a scalable and responsive web application. Because of its focus on teamwork as well as individual productivity, its design is adaptable to a wide range of use cases.

Figure 1.1: Sample of the Output

Chapter 2

METHODOLGY

2.1 Research methodology/Tool used

A succinct process for studying and designing a To-Do List application consists of three important stages: requirement analysis, which comes first, uses surveys and market research to determine the major features and user demands. The system architecture and user interface are carefully developed throughout the design process that follows, guaranteeing a balance between functionality and user experience. During the development phase, the application is constructed with the help of backend technologies like Node.js and maybe a front-end framework for a responsive user interface. To guarantee dependability and customer happiness, testing is essential and includes unit, integration, and user acceptability testing. The program is deployed to a server or cloud platform after testing has completed successfully, enabling end users to access it. It is imperative to identify areas for improvement post-deployment, through evaluation and input from real users, as this will result in iterative changes throughout the ensuing maintenance phase.

2.1.1 About Node.js and Express.js

Developers can run JavaScript server-side with Node.js, an open-source, cross-platform JavaScript runtime environment. Because Node.js is based on the same V8 JavaScript engine as Google Chrome, it can execute JavaScript code with remarkable speed even when used outside of a web browser. Because JavaScript can now be used on both the client and server sides, this feature has completely changed web development and made it easier to create scalable, quick network applications.

Express.js, also commonly known as Express, is a feature-rich Node.js web application framework that is simple to use and offers a wide range of functionalities for creating both web and mobile applications. By offering middleware and utility methods, it speeds up the creation of server-side applications and enables programmers to build safe, modular, and maintainable code. Express.js is

the de facto standard for Node.js web applications because it makes the server construction process that is necessary for Node.js simpler and offers crucial features like routing, templating, and handling requests and answers.

Node.js and Express.js work together to create a potent combination for backend development that makes creating web apps, APIs, and microservices quick and easy. They are especially well-suited for applications that demand high concurrency and real-time capabilities, including chat programs and live updates, because of their non-blocking, event-driven architecture. The vast npm (Node Package Manager) ecosystem, which gives access to thousands of reusable packages and speeds up development even more, has contributed to the broad adoption of Node.js and Express.js.

2.1.2 About Custom vision

This To-Do List project's customized vision goes beyond task management's fundamental features to provide a distinctive, user-centric experience catered to certain requirements and lifestyles. Fundamentally, it seeks to minimize stress and increase productivity through tailored user experiences, smooth integration with current tools and platforms, and intelligent job prioritization. By focusing on a particular group of people, such as working families, students, or professionals, the initiative hopes to offer a solution that addresses the particular difficulties that these groups confront.

The integration of cutting-edge technologies like intelligent prioritizing algorithms, AI-driven job optimization recommendations, and an incredibly configurable user interface that enables users to exactly tailor the application to their workflow is essential to realizing this vision. In addition, the program is made with future expansion and scalability in mind, so it may change to meet the changing needs of its users. The project is not merely a tool for managing to-do lists; rather, it is a partner in attaining a balanced, productive life. It goes beyond simple task management by striving to promote a sense of accomplishment and well-being among its users. This project's unique vision promises a To-Do List application that will not only be functional but also genuinely revolutionize how people manage their time and chores.

2.2.2 Step-By-Step Procedure

In our project, numerous procedures are involved and within those procedures, various small steps are also involved. The procedure are as follows:

1) Setting Up Your Node.js Environment

Install Node.js and npm:

Get the operating system installer by going to the official Node.js website. Node.js and npm (Node Package Manager), which is used to manage dependencies for Node.js applications, are both included in the installer.
Launch a Novel Node.js Project:

Get your command prompt or terminal open.
Go to the directory where your To-Do List project is to be created.
Execute npm init. You will be prompted to enter numerous project-related details (such as the name, version, and description) when you run this program. You can click enter to accept the settings or fill these up. Your project's dependencies and other configurations are tracked in a package.json file that is created in your project directory by this procedure.

2) Installing Express

3) Install Express with npm:
Run npm install express in the same directory as your package.json file. By adding Express and its dependencies to your project and downloading them, you may update the package.json file to reflect the new dependency.Express is a Node.js web framework that makes it simpler to create and route servers, which facilitates the development of web applications and APIs.

3)Set Up Your Server Using Express:

- Launch your text editor and open app.js Add const express = require('express'); to the top of the file to require Express.

- Incorporate the expression app by adding const app = express(); to create an instance.

- Give your server a port to listen on, such as const PORT = 3000;.
- b. Add app to instruct your Express app to listen on the designated port.listen(PORT,()) => console.log(${PORT}) when the server is running

4)Define Routes:

To design routes for showing the to-do list, adding new tasks, and deleting tasks, use the app.get, app.post, and app.delete methods, accordingly. Every route shall possess a callback function that specifies the actions to be taken when accessing the route.

5)Setting Up the Frontend

- Create a basic user interface (UI): b) Generate HTML files for your website's structure. An input form for new tasks, a submit button, and a list displaying the activities are possible components of a to-do list.

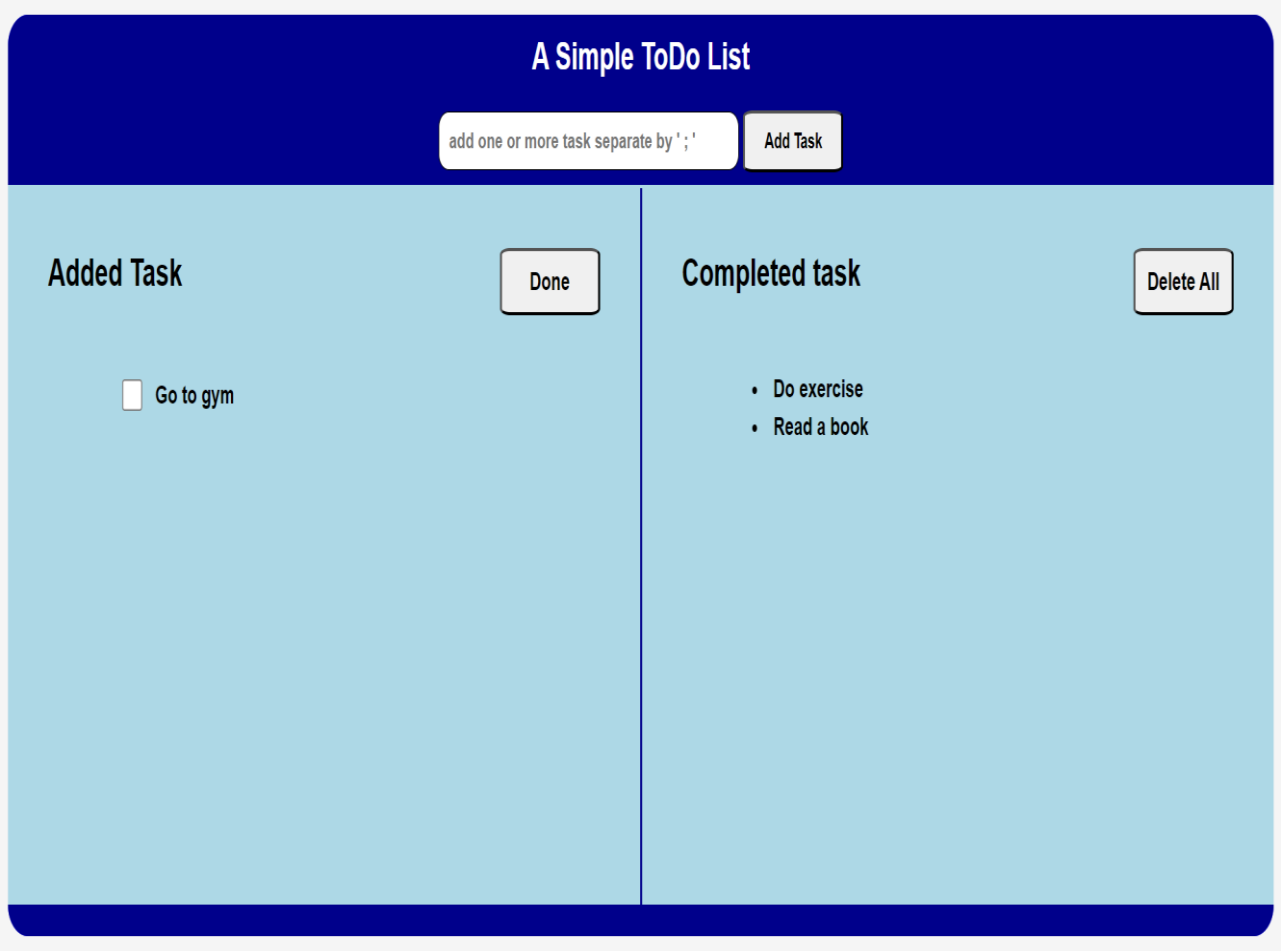
- Style your pages with CSS files. Styles can be specified for your input field, list, buttons, and other elements.

- Use JavaScript to write code that performs dynamic actions, such adding tasks to a list without refreshing the browser.

4)Serve Your Static Files:

- Make a public folder in the project directory. You should save your static files (HTML, CSS, JS) here.
- Add app.use(express.static('public')); to your app.js file to instruct Express to serve static files from the public directory. This piece of code, which opens the contents of the public folder to the web, must come before your route declarations.
- These instructions let you install and setup Express to build a server, set up a basic frontend to communicate with your server, and set up a minimal Node.js environment for your to-do list application. As you get more familiar with Node.js and Express, you may build upon this configuration by adding more features and functionalities.

OUTPUT:


A Simple ToDo List
add one or more task separate by ' ; '
Add Task
Added Task
Done
Go to gym
Completed task
Delete All
Do exercise
Read a book

Chapter 3

Implementation

1.1 Real world applications

To-do lists are flexible tools that may be used in a wide range of real-world contexts to improve productivity and organization in both personal and professional contexts. They are essential for project management in the workplace because they let teams divide large, complicated projects into smaller, more manageable tasks, rank those tasks according to significance and urgency, and monitor their progress.

To-do lists help people organize their days and handle both personal and professional obligations, such as important business meetings and family duties. They are also very important in classrooms, where they are used by teachers and students to set up study plans, keep track of due dates for assignments, and schedule revision sessions.

To-do lists have developed beyond its conventional applications to assist goal-setting and personal growth endeavors, allowing users to create, monitor, and accomplish both short- and long-term objectives. To-do lists are fundamental tools that help us organize our time, complete our jobs more quickly, and free up our minds so that we can concentrate on the things that really important in both our personal and professional lives.

Chapter 4

Conclusion and Future work

4.1 Conclusion

To sum up, the To-Do List project, which makes use of Node.js and Express.js, is a big step in the right direction toward improving user productivity and task management. Through the combination of Express.js's ease of use and flexibility with Node.js' powerful backend capabilities, the project creates a scalable, user-friendly application that caters to a wide range of audience needs. It stands out in the crowded productivity tool market thanks to its unique vision, which is centered on offering a customized and simple user experience. With cutting-edge features like intelligent prioritization, smooth integrations, and AI-driven recommendations, it provides more than simply a work management tool—rather, it acts as a partner in accomplishing both personal and professional objectives. This project's real-world applications span multiple sectors, demonstrating its adaptability and usefulness in managing both short-term and long-term goals. Overall, this To-Do List project demonstrates the transformative power of carefully crafted software solutions in enhancing productivity and organization in our lives, in addition to showcasing the technical ability of Node.js and Express.js.

4.2 Future work

In order to improve its usefulness and user involvement, this To-Do List project will go in a number of fascinating areas in the future. In order to deliver more personalized and predictive task management capabilities, like automatic task categorization, prioritization based on user behavior, and intelligent reminders, one main focus will be on integrating advanced AI and machine learning algorithms. To further increase the app's usefulness in multi-tool workflows, its integration possibilities with other productivity tools and platforms should be expanded. This will guarantee that users can synchronize their operations across various contexts with ease. In order to meet the demands of professional settings, collaboration tools that allow users to share lists, assign tasks, and monitor progress in team projects will be included as well.

Chapter 5

References

1. Youtube

   https://www.youtube.com/watch?v=G0jO8kUrg-I&pp=ygUlc2ltcGxlIHRvZG8gbGlzdCBwcm9qZWN0IHVzaW5nIG5vZGVqcw%3D%3D

2. Various websites

   https://www.geeksforgeeks.org/how-to-make-to-do-list-using-nodejs/

   https://github.com/topics/todo-nodejs

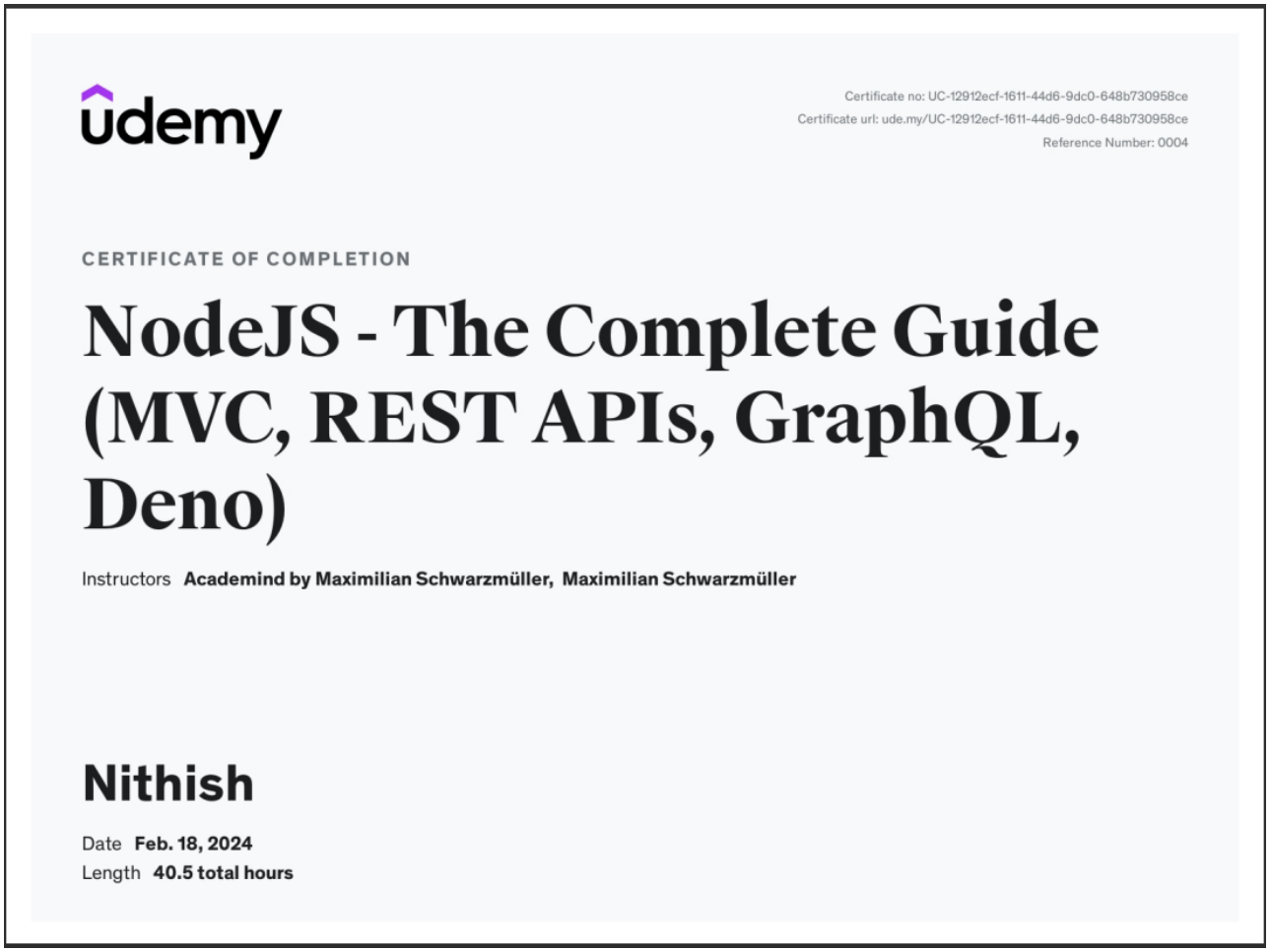
   https://support.glitch.com/t/how-to-create-a-todo-list-in-node-js/65098

   https://www.tatvasoft.com/blog/node-js-best-practices/

   https://www.freecodecamp.org/news/how-to-build-a-todo-app-with-react-typescript-nodejs-and-mongodb/

Chapter 6

Certification


udemy

Certificate no: UC-12912ecf-1611-44d6-9dc0-648b730958ce
Certificate url: ude.my/UC-12912ecf-1611-44d6-9dc0-648b730958ce
Reference Number: 0004

CERTIFICATE OF COMPLETION

NodeJS - The Complete Guide (MVC, REST APIs, GraphQL, Deno)

Instructors **Academind by Maximilian Schwarzmüller, Maximilian Schwarzmüller**

Nithish

Date **Feb. 18, 2024**
Length **40.5 total hours**

Figure 6.1: Certification details