Full-Stack Task Management Application Documentation

Overview

This document provides an overview of the Full-Stack Task Management Application, detailing the features, technologies used, and implementation steps.

To RUN this project Both (FE and BE)

```
npm install

npm run dev
```

Features

- Display a task list with details such as title, description, and status.
- Allow users to upload Excel files containing tasks.
- Implement CRUD operations for tasks (Create, Read, Update, Delete).
- Attach PDF files to tasks.
- Filter tasks based on completion status.

Technologies Used

- HTML5
- CSS3
- JavaScript (ES6+)
- vueJS
- Node.js
- Express.js
- Multer (for file upload handling)
- XLSX (for Excel file parsing used) (---(xlsx *

Severity: high

Prototype Pollution in sheetJS - https://github.com/advisories/GHSA-4r6h-8v6p-xvw6

No fix available

node_modules/xlsx)

)

- Sequelize (for ORM in Node.js)
- RESTful API design principles
- Fetch API
- Callback functions
- Async/Await
- PDF file handling

Implementation

HTML & CSS

- Created an HTML page to display the task list.
- Styled the task list using CSS to ensure a visually appealing and user-friendly interface.

JavaScript (ES6+)

- Utilized destructuring and template literals for task information extraction and summary generation.
- Implemented asynchronous functions using async/await to fetch tasks from a REST API endpoint.

- Utilized Fetch API for making HTTP requests.

Node.js & File Handling

- Set up an Express.js server.
- Integrated Multer middleware to handle file uploads.
- Created routes for handling GET requests to fetch tasks and POST requests to upload Excel files.
- Extended server routes to handle Excel files and used the XLSX library to parse them.

REST API & Callback Functions

- Designed REST API endpoints for listing, creating, updating, and deleting tasks.
- Implemented a callback function to filter completed tasks from a list fetched from the API.

PDF File Handling

- Updated the task data structure to include file paths for attached PDFs.
- Modified HTML to display links to attached PDF files for each task.

Full Stack Integration

- Combined VueJS, HTML, CSS, JavaScript, Node.js, Excel file handling, REST API, callback functions, and async/await concepts to create a full-stack task management application.
- Implemented features for adding, updating, and deleting tasks, including attaching and viewing PDF files and uploading tasks from an Excel file.

Conclusion

The Full-Stack Task Management Application provides a comprehensive solution for managing tasks efficiently. By leveraging a variety of technologies and implementing key features, the application offers users a seamless experience for task management, file handling, and data manipulation.