Software Documentation

Prepared by

N.K. Mallikarachchi

April 2024

Software Documentation

Software Name:	Futute ToDo
Date:	10/04/2024
Version:	1.0
By:	N.K. Mallikarachchi

Version History:

Version	Author	Updated Date
1.0	N.K. Mallikarachchi	10/04/2024

Review History:

Version	Author	Updated Date
1.0	N.K. Mallikarachchi	10/04/2024

Contents

List of Figures

List of Tables

1 Introduction

In software development process documentation serves as the backbone of understanding, communication and maintenance. This document serves as a guide for developers, testers, and other stakeholders involved in the development and usage of To-do application named Future ToDo. Future ToDo is a web-based task management application. This ensures clarity, efficiency and reliability throughout the application's lifecycle.

1.1 Purpose

The purpose of this application is to provide users with a simple effective tool for managing their daily tasks. With the use of create, read, update and delete To-do items the users can easily organize their workload and stay productive.

1.2 Scope

This documentation covers both the frontend and backend aspects of the application. On the backend, we have developed a RESTful API using C# .NET which interacts with SQLite database to manage to-do items. The frontend is built using ReactJS and TypeScript to provide better user experience.

1.3 Audience

This documentation is intended for:

- **Developers**   : Who will be involved in the development and maintenance of the application.
- **Testers**: Who will be responsible for testing the application's functionality.
- **Project Managers**: Who will oversee the development process and ensure project milestones are met.
- **End Users**: Who will interact with the application to manage their to-do items.

2 Getting Started

This project was created with Create React App.

2.1 Clone the Project

Cloen the repository using "gh repo clone Nipunikumudika/ToDoWeb-Assignment" in github CLI. Or you can go to "https://github.com/Nipunikumudika/ToDoWeb-Assignment.git" using your internet browser and go to Code and Download as zip and extract.

2.2 Available Scripts

2.2.1 For Running Backend

Open a new command window in project directory and run

cd .\backend\API\ToDoBackend

Then run

Dotnet run

2.2.2 For running Frontend

Once you clone the project you can go to frontend folder using command line.

cd .\frontend

Then run **npm i** to install pacages

Then, you can run:

npm start

This will run the app in the development mode.

Open http://localhost:3000 to view it in the browser.

And you will also see any lint errors in the console and as toasts.

npm test

This launches the test runner in the interactive watch mode.

npm run build

Builds the app for production to the build folder. It correctly bundles React in production mode and optimizes the build for the best performance. The build is minified and the filenames include the hashes. Then the app is ready to be deployed!

npm run eject

Note: this is a one-way operation. Once you eject, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can eject at any time. This command will remove the single build dependency from your project.

This will run both frontend and backend. Here make sure that port 3000 and 7110 not used in any other program.

You will see below type of UI in desktops or laptops when successfully loading the frontend. Note that the UI in smaller screens will slightly different from this.
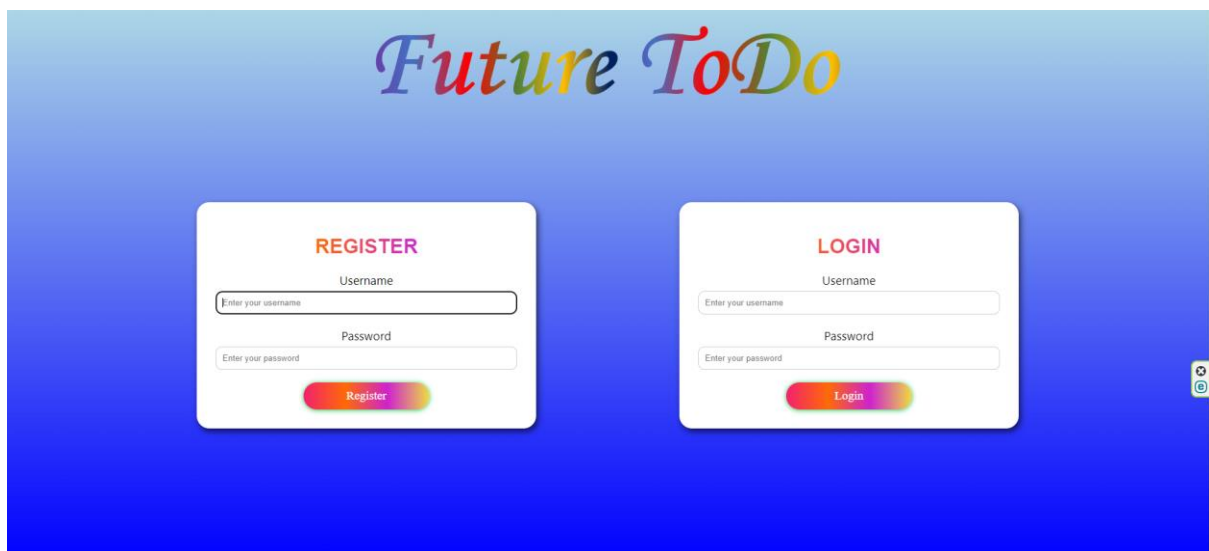
Figure 2-1: Main Screen

3 User Guide

In the main page you can register or login to the system using username and password. When registering you are not allowed to use any username that are priory used by our users.

 If you enter that type of username you will get a toast message saying "Error"
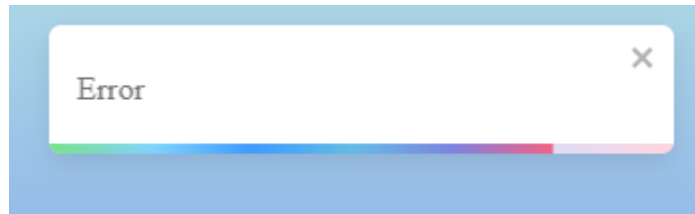
Figure 3-1: Registration Error Message

If your username or password is incorrect you will see a toast message saying "Check Username & Password"
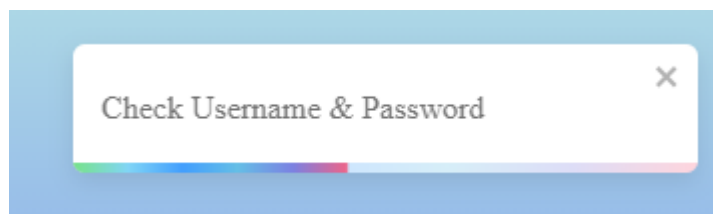
Figure 3-2: Wrong Username or Password Error Message

If you successfully login to the system at the first time you will see a page like below.
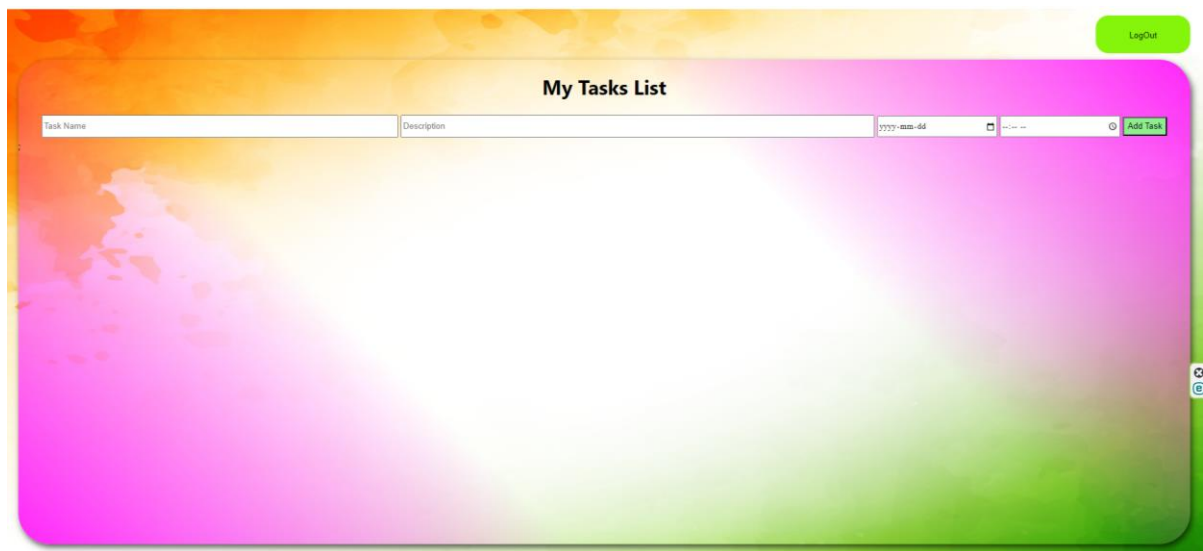
Figure 3-3: Task List Page for New User

3.1   Add a task

You can add a task using below section

Figure 3-4: Adding g a Task

In here make sure that you fill all the boxes and date and time is greater than the time of you add the task. If not, you will prompt error messages.

Figure 3-5: Not filling all Detail Error Message

Figure 3-6: Date and Time not Greater than Now Error Message

After adding tasks user UI will look like below
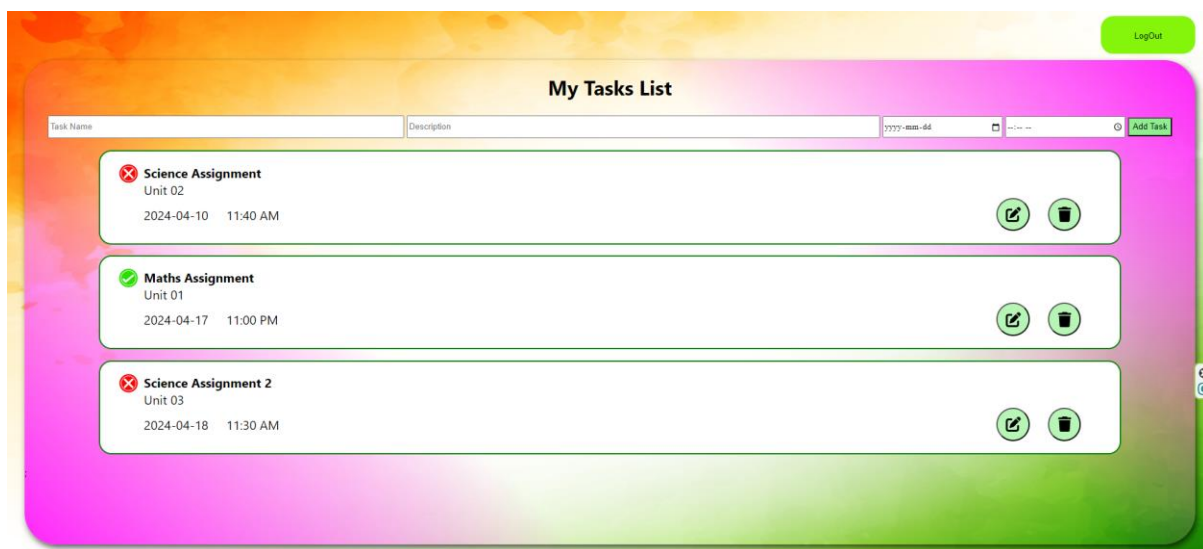
Figure 3-7: User Interface After Adding Tasks

3.2 Edit a Task

In here you can edit the tasks and make them completed by clicking edit button

Figure 3-8: Edit Button

Then you will see a window like below. There you can change all the details and click change button. Make sure that the date you enter is greater than the time you fill this. If it is not greater you can change the status only to completed. If not, you will get an error toast message.
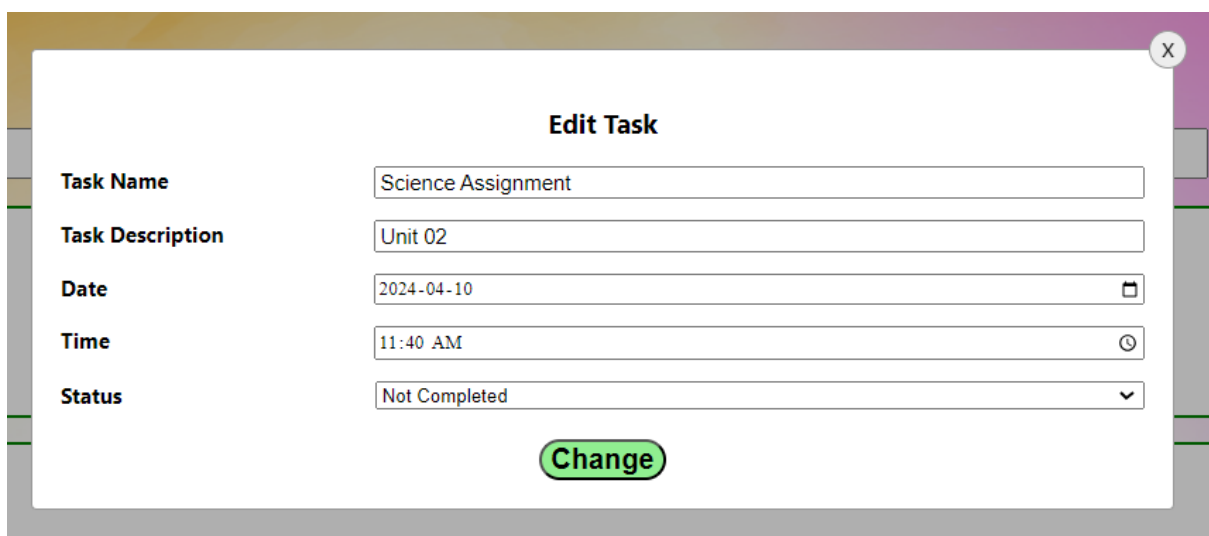
Figure 3-9: Edit Button Window

If the date and time are over and if you still did not do the task you will see it in UI like below.

Figure 3-10: Not Completed Task

If you completed task and the date and time is over you will see it in UI like below.

Figure 3-11: Completed Task and Time is Over

If the task is not completed, you will see below icon at the start of you task.

Figure 3-12: Not Completed Task Icon

If the task is completed, you will see below icon at the start of you task.

Figure 3-13: Completed Task Icon

3.3 Delete a Task

You can delete a task by clicking delete icon. If successful, you will get a toast message saying that "Deleted".

Figure 3-14: Delete Button

Figure 3-15: Successfully Delete a Task

3.4 Logout

You can logout from the system by clicking bellow button. It will send you to the login and register page again.
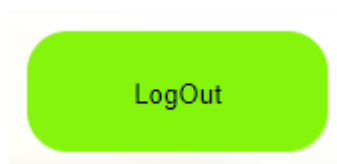
Figure 3-16: Logout Button

4 API Endpoints
4.1 Create a User

HTTP Method   : POST

Endpoint   : /api/Users

Description   : Allows creating a new user.

Request Body   : Username (string): The username of the new user.

                            Password (string): The password of the new user.

Response   : 200 OK: Returns a list of all users after successful creation.

                            400 Bad Request: If the request body is null.

4.2 Login

HTTP Method   : POST

Endpoint   : /api/Users/login

Description   : Allows user authentication and login.

Request Body   : Username (string): The username of the user trying to login.

                            Password (string): The password of the user trying to login.

Response   : 200 OK: Returns the user object upon successful login.

                            401 Unauthorized: If the user does not exist or the password does not match.

4.3 Get All Tasks

HTTP Method   : GET

Endpoint   : /api/Tasks

Description   : Retrieves all tasks.

Response   : 200 OK: Returns a list of all tasks.

                            404 Not Found: If no tasks are found.

4.4 Get Task by ID

HTTP Method   : GET

Endpoint   : /api/Tasks/{id}

Description   : Retrieves a task by its ID.

Parameters   : {id} (int): The ID of the task to retrieve.

Response   : 200 OK: Returns the task object.

                         404 Not Found: If the task with the specified ID is not found.

4.5 Create a Task

HTTP Method   : POST

Endpoint   : /api/Tasks

Description   : Creates a new task.

Request Body   : TaskName (string): The name of the task.

                         Description (string): The description of the task.

                         Date (DateTime): The date of the task.

                         Time (TimeSpan): The time of the task.

                         TaskStatus (string): The status of the task.

Response   : 200 OK: Returns a list of all tasks after successful creation.

                         400 Bad Request: If the request body is null.

4.6 Delete a Task

HTTP Method   : DELETE

Endpoint   : /api/Tasks/{id}

Description   : Deletes a task by its ID.

Parameters   : {id} (int): The ID of the task to delete.

Response   : 200 OK: Returns a list of all tasks after successful deletion.

                              404 Not Found: If the task with the specified ID is not found.

13

4.7 Update a Task

HTTP Method   : PUT

Endpoint   : /api/Tasks/{id}

Description   : Updates a task by its ID.

Parameters   : {id} (int): The ID of the task to update.

Request Body   : TaskName (string): The name of the task.

                              Description (string): The description of the task.

                              Date (DateTime): The date of the task.

                              Time (TimeSpan): The time of the task.

                              TaskStatus (string): The status of the task.

Response   : 200 OK: Returns the updated task object.

                              404 Not Found: If the task with the specified ID is not found.

                              400 Bad Request: If the request body is null.

5   Additional Information

5.1   Calling APIs

Normally this uses port 7110. So you can call above API endpoints using below example URL. Make sure to change URL according to user needs.

https://localhost:7110/api/Users

5.2   Database Location

Sqlite database is located in below position.

\ToDoWeb-Assignment-main\backend\API\ToDoBackend\sqlite.db

You can use DB Browser (Sqlite) to open this database.

5.3   Software Versions

When creating our project we use below softwares and versions.

5.3.1   Backend

Table 1: Backend Software Versions

Name	Version
Microsoft.EntityFrameworkCore	6.0.28
Microsoft.EntityFrameworkCore.Design	6.0.28
Microsoft.EntityFrameworkCore.Sqlite	6.0.28
Microsoft.EntityFrameworkCore.Tools	6.0.28
Swashbuckle.AspNetCore	6.2.3

5.3.2 Backend

Table 2: Frontend Software Version

Dependency	Version
@fortawesome/fontawesome-svg-core	6.5.2
@fortawesome/free-solid-svg-icons	6.5.2
@fortawesome/react-fontawesome	0.2.0
@testing-library/jest-dom	5.17.0
@testing-library/react	13.4.0
@testing-library/user-event	13.5.0
@types/jest	27.5.2
@types/node	16.18.94
@types/react	18.2.74
@types/react-dom	18.2.24
axios	1.6.8
bcryptjs	2.4.3
crypto-browserify	3.12.0
react	18.2.0
react-dom	18.2.0
react-router-dom	6.22.3
react-scripts	5.0.1
react-toastify	10.0.5
typescript	4.9.5
web-vitals	2.1.4