aifutuerx_task

Prepared by: Bilal Khasawneh

---

1. Project Overview

This document provides a detailed overview of the implementation, challenges faced, solutions applied, and total hours spent on the **aifutuerx_Task** project. The project was developed using **Angular** (frontend) and **.NET Core Web API** (backend) and integrates with a **SQL Server database**.

The main objective of the project was to create a **robust, secure, and user-friendly task management system** with features like task management, export functionality, authentication, and responsive UI.

---

2. Implementation Details

- **Backend (C# .NET Core API):**
  - Developed RESTful API endpoints for task CRUD operations.
  - Implemented **JWT-based authentication and authorization** for secure access.
  - Built **password reset with email OTP verification** to enhance account security.
  - Integrated the backend with SQL Server and ensured proper validation for data consistency.
- **Frontend (Angular):**
  - Designed a **responsive task table interface** for managing tasks efficiently.
  - Implemented **dark mode support** for improved user experience.
  - Added **export to PDF functionality** for easy reporting and record-keeping.
  - Integrated Angular frontend with the .NET Core API for seamless data communication.
- **Additional Features:**
  - Real-time form validation and error handling for user inputs.
  - Modular and maintainable code structure to facilitate future scalability.
  - Optimized loading performance for faster user interaction.

---

3. Total Hours Spent

The project was completed in approximately **5 hours** of development spread across multiple sessions.

4. Challenges and Solutions

Challenge	Solution
Integrating **JWT authentication** with Angular frontend	Followed official documentation and applied token storage in **localStorage** with proper guards for routes.
Implementing **OTP-based password reset**	Researched email services and implemented **time-limited OTP** caching with proper validation.
Generating **PDF export functionality** in Angular	Used **jsPDF library**, tested different formats, and ensured table responsiveness in PDF output.
Handling **dark mode toggle** persistently	Used **Angular services** and CSS variables to maintain user preference across sessions.

By addressing these challenges using official documentation, best practices, and online resources, the project maintains **robust security, reliability, and user-friendly experience**.

---

5. Best Practices Followed

- Used **modular code architecture** in both Angular and .NET Core for maintainability.
- Applied **secure coding practices** to prevent common vulnerabilities like SQL injection and token theft.
- Ensured **responsive UI design** to support multiple device types.
- Maintained **clear documentation** for future reference and team collaboration.

---

6. Conclusion

The result is a **fully functional, secure, and user-friendly task management application** integrating Angular frontend with .NET Core backend. This project demonstrates proficiency in modern web development practices and adherence to security and performance standards.

I hope this work meets your expectations, and I look forward to **joining your team, learning from your expertise, and contributing to delivering excellent solutions**.

Prepared by:
Bilal Khasawneh