Documentation: NWU Taskmaster
CMPG 323
24904635
F.P. Burger

Introduction

NWU Taskmaster is a web application designed to streamline tasks and reminder management. With user authentication, category-based task organization, and timely notifications, it ensures an efficient workflow for users.

To use the application you'd simply need to visit the website at https://nwu-taskmaster.azurewebsites.net
However if you wish to edit the site or add to it you'd need to follow this simple setup guide.

Setup

1. **Clone the Repository**
   Clone the GitHub repository to your local system.
2. **Install Dependencies**
   - Open the solution in **Visual Studio**.
   - Restore NuGet packages automatically when opening the solution.
3. **Database Setup**
   - **Azure SQL Database**: Create a new database using the Azure Portal.
   - **Local Database**: Use SQL Server Management Studio (SSMS) or the SQL Server Object Explorer in Visual Studio.
   - Run the provided CreateTables.sql scripts to initialize the database structure. Can be found in the Scripts folder for each table.
   - Update the connection string in Web.config to match your database connection:
4. **Publish to Azure** (Optional)

    - Use the **Azure App Service** for hosting.
    - Deploy via Visual Studio or GitHub Actions.

5. **Run the Application**
    - Set the project as the startup project in Visual Studio.
    - Start the application using Ctrl + F5 or click the **Start** button.

Usage

1. **Authentication**
    - Users must **register** or **log in** to access the system.
    - Only authenticated users can manage tasks and reminders.
2. **Dashboard**
    - Displays all tasks in chronological order.
    - Features include:
        - **Add Tasks**: Create new tasks with details like due date, priority, and category.
        - **Edit Tasks**: Modify existing tasks.
        - **Delete Tasks**: Remove tasks no longer needed.
        - **Filter by Category**: View tasks by their assigned category.
3. **Reminders**
    - Set reminders for tasks with specific dates and times.
    - Notifications appear in the navigation bar when reminders are due.
4. **Task Categories**
    - Categories are dynamic and customizable.
    - Tasks can be filtered by category for easy navigation.
5. **Reminders Management**
    - Access reminders through the **Reminders** menu.
    - Add reminders directly from the reminders page and delete reminders through the Reminders menu.

Development Notes

1. **Technologies**
   - **ASP.NET MVC 4.8**: Framework for application development.
   - **SQL Server**: Database backend for data storage.
   - **Bootstrap 5**: Frontend framework for responsive design.
2. **File Structure**

Controllers

Handles application logic and manages user requests. Key controllers include:

- **AccountController.cs**: Manages user authentication, registration, and profile-related actions.
- **DashboardController.cs**: Handles logic for displaying and filtering tasks on the dashboard.
- **HomeController.cs**: Controls general navigation, including the homepage.
- **ManageController.cs**: Handles account and profile management tasks.
- **RemindersController.cs**: Manages reminders and notifications for tasks.
- **TaskController.cs**: Manages CRUD operations for tasks, including subtasks.

Models

Defines the structure and relationships of data entities:

- **AccountViewModels.cs**: Represents user registration, login, and account management data.
- **Clients.cs**: Links users to their associated tasks and reminders.
- **ClientTasks.cs**: Represents individual tasks created by users.
- **IdentityModels.cs**: Handles ASP.NET Identity-related data models for user authentication.
- **ManageViewModels.cs**: Provides models for account and password management.
- **Reminders.cs**: Represents reminders associated with tasks.
- **ReminderViewModels.cs**: Provides view-specific models for creating and managing reminders.
- **TaskCategories.cs**: Represents categories to organize tasks.
- **TaskViewModels.cs**: Provides view-specific models for task creation and editing.

Services

Contains reusable business logic and utility classes:

- **ReminderService.cs**: Manages logic for triggering and notifying reminders.

Views

Razor views that render user interfaces for various features:

- **Account/**: Contains views for user authentication and account management (e.g., Login, Register).
- **Dashboard/Index.cshtml**: Displays tasks and categories with filtering functionality.
- **Home/**: Contains the landing page and navigation.
- **Manage/**: Includes views for user profile and password management.
- **Reminders/Index.cshtml**: Allows users to create, manage, and view reminders.
- **Shared/**: Contains shared layouts and partial views (e.g., _Layout.cshtml for consistent site-wide navigation).
- **Task/Index.cshtml**: Provides task creation and editing forms.

Other Resources

- **Site.css**: Contains custom CSS for styling the application's UI.

3. **Background Services**
   - A **Task Scheduler** is implemented to monitor reminders periodically.
   - Reminders are dynamically fetched and displayed in the navigation bar.
4. **Responsive Design**
   - The application is fully responsive, ensuring usability across desktops, tablets, and mobile devices.

Troubleshooting

1. **Common Errors**
   - **"Database not found"**: Verify that the connection string in Web.config matches your database configuration.
   - **Authentication Issues**: Ensure cookie expiration is properly configured in Startup.Auth.cs.
2. **Performance Tips**
   - Regularly clean up old reminders and tasks to optimize database performance.

   - Avoid storing large files or images directly in the application. Use cloud storage for such purposes.

3. **Debugging Background Tasks**
   - Check the logs to ensure the Task Scheduler is running correctly.
   - Verify reminders are set with valid dates and times.

Future Enhancements

1. **Email Notifications**: Enable email-based reminders for tasks.
2. **Calendar View**: Add a visual calendar for better task organization.
3. **Edit Reminders**: Update the Reminders page with the ability to edit the reminders.
4. **Subtask Capability**: Originally planned, but did not come to fruition, the option to add subtasks to an existing task as to split bigger projects into more manageable goals.
