TFB1033: OBJECT ORIENTED PROGRAMMING

SEPTEMBER 2024 SEMESTER

PROJECT REPORT:
WORK SHIFT SCHEDULER

PREPARED BY:

CodeForBiz

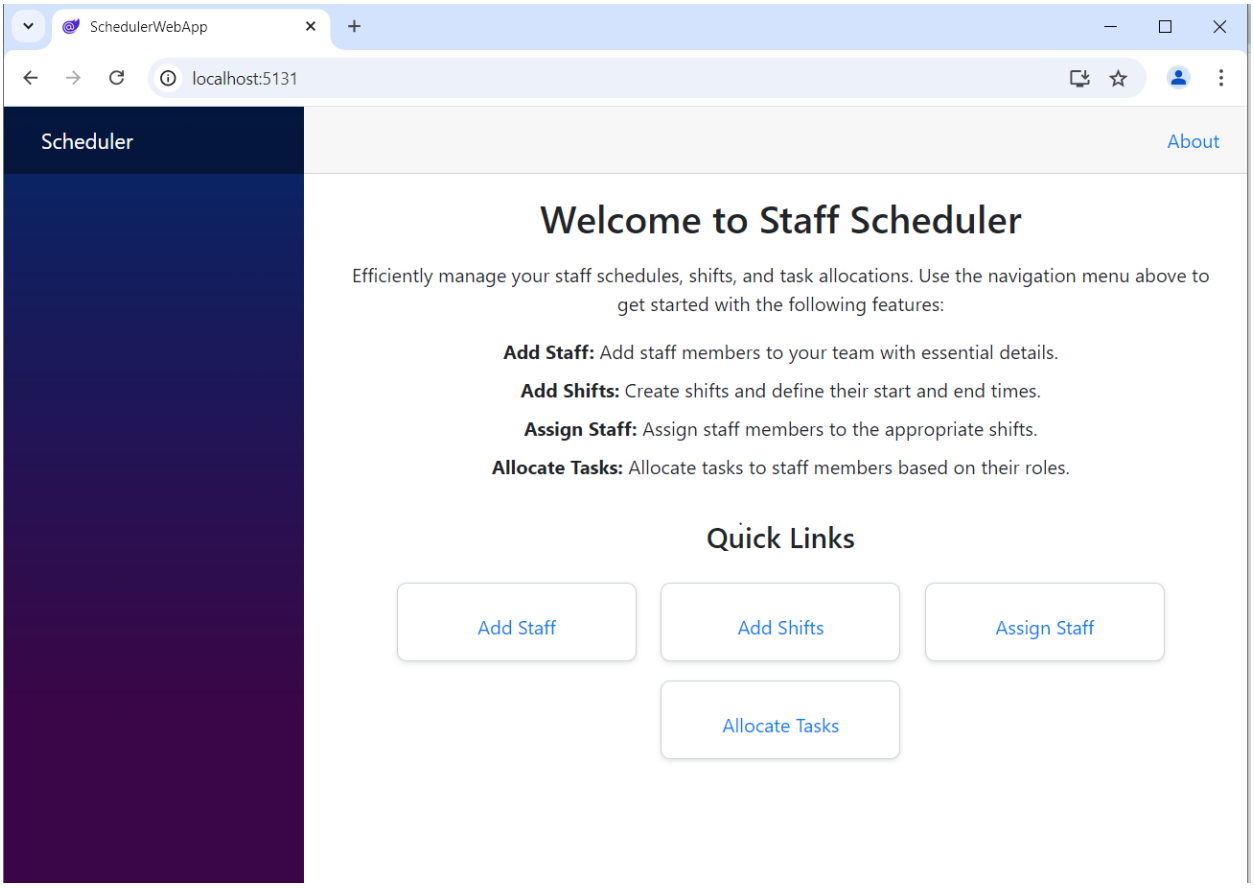
STUDENT NAME	STUDENT ID
NURUL SYAHIRA BINTI ABDUL MUGHNII	22005582
MOHAMAD ADAM BIN MOHD FAIZAL	22002016
DASHITA VADIVEL	22010193
MUHAMMAD KHIR AKASHAH BIN KHAIRUL EHSAN	22006614
NUR ALYA JAZMINA BINTI KHAIRUL AZHAR	22005061

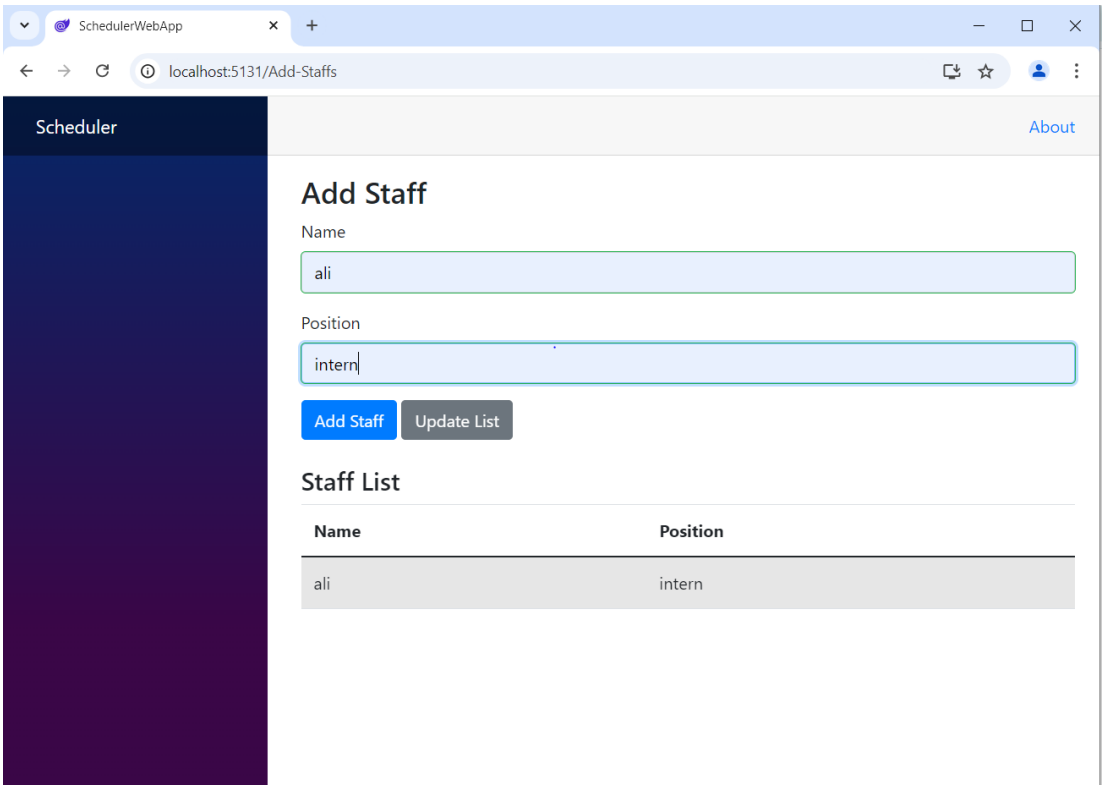
Contents

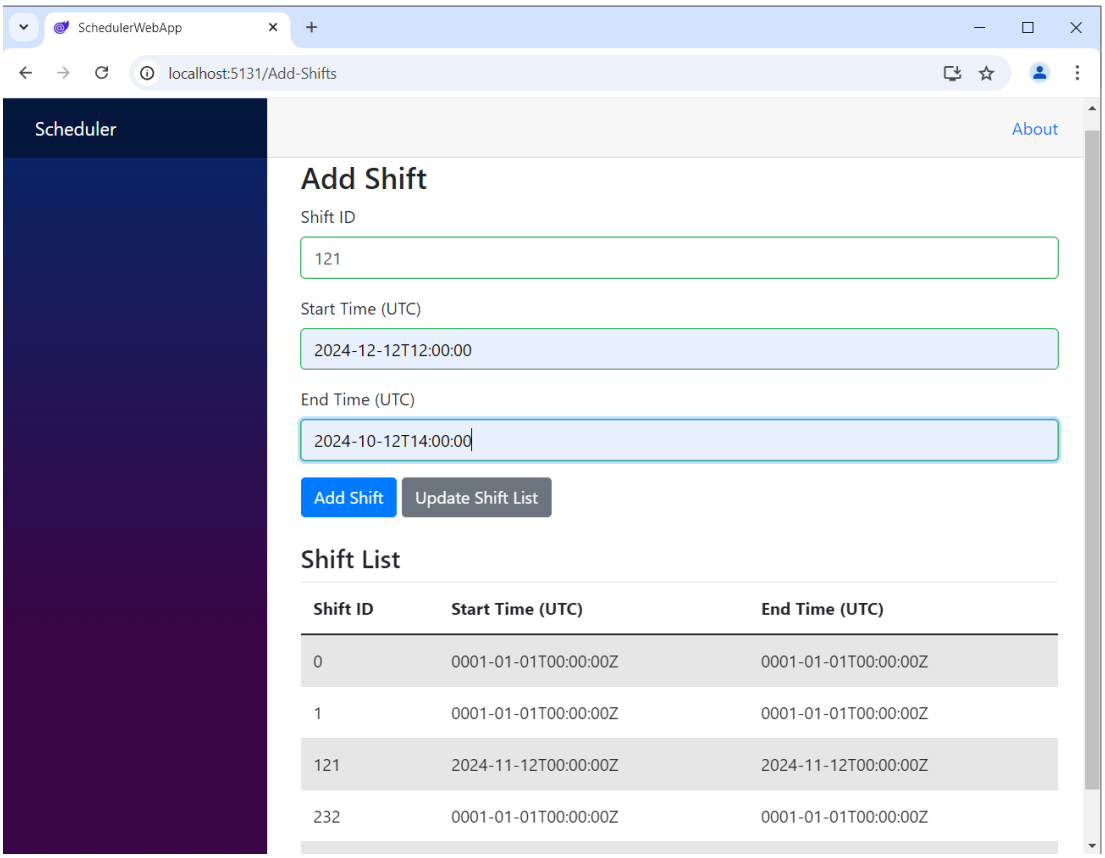
1.0 PROJECT DESCRIPTION

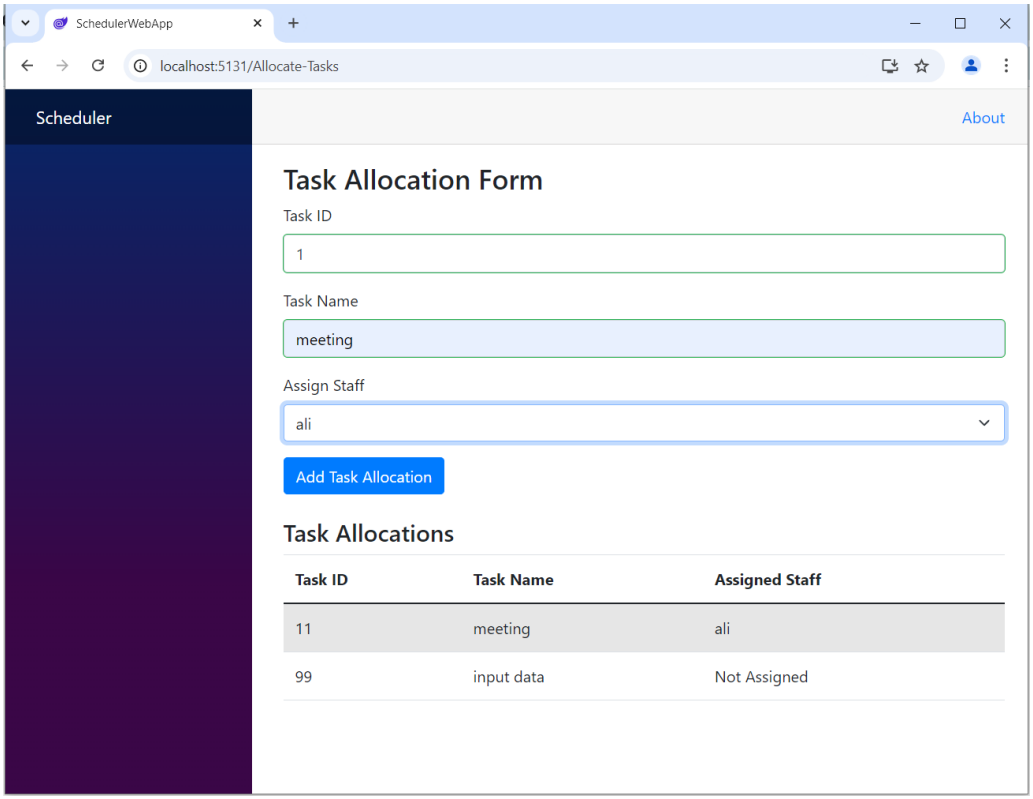
Our staff scheduling and shift management application is designed to simplify shift planning for small businesses. With an easy-to-use interface, managers can effortlessly organize employee shift timings, assign tasks, and manage schedules with just a few clicks, while employees can view their shifts, request time off, or swap shifts in real-time, improving communication and flexibility. This application is especially useful for small businesses as it reduces the time spent on manual scheduling, prevents shift conflicts, and ensures proper staffing levels, helping managers focus more on growing their business instead of administrative tasks. Best of all, the application is completely free with no subscription fees, offering small businesses a valuable tool for managing their workforce without any financial burden.

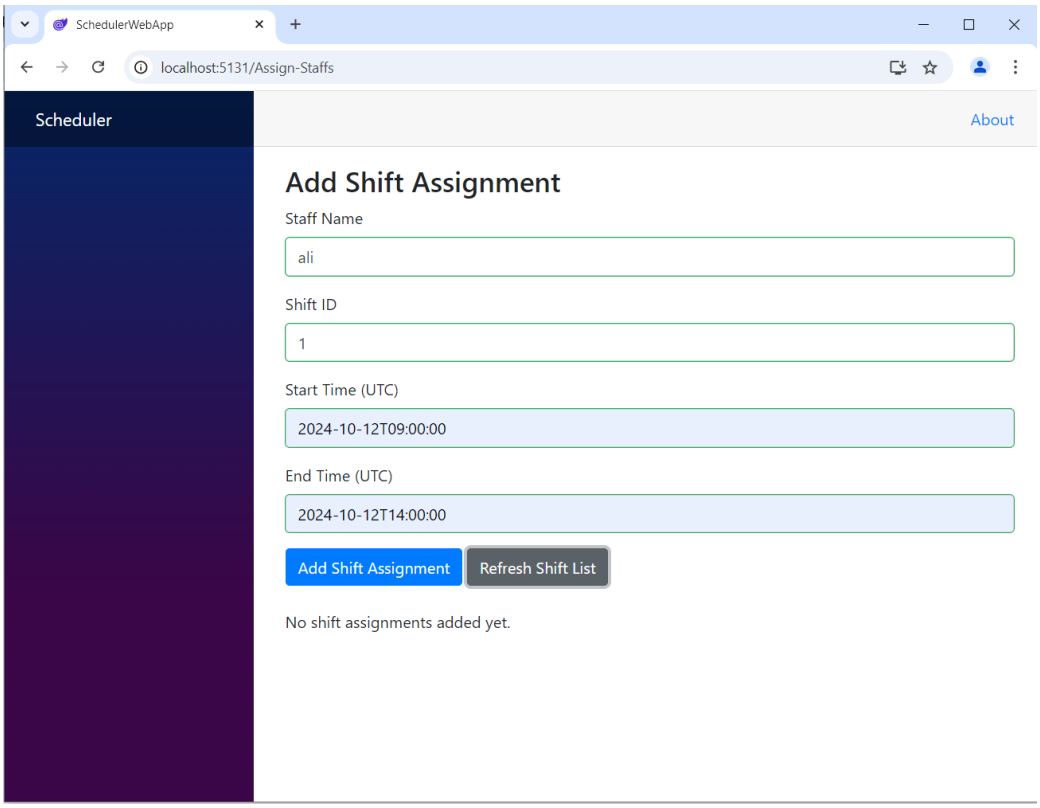
2.0 SCREENSHOT



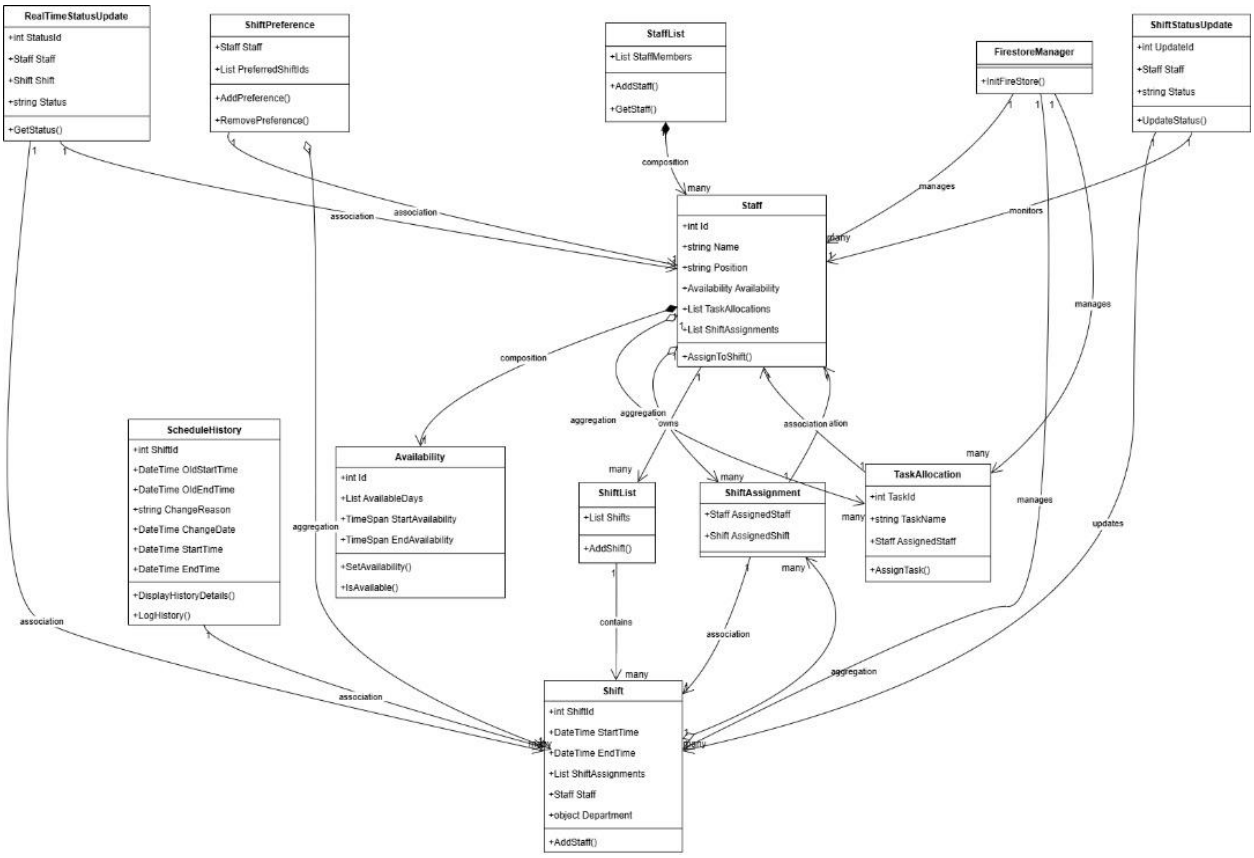

SchedulerWebApp
localhost:5131/Add-Staffs
Scheduler
About
Add Staff
Name
ali
Position
intern
Add Staff
Update List
Staff List
Name
Position
ali
intern


SchedulerWebApp
localhost:5131/Add-Shifts
Scheduler
About
Add Shift
Shift ID
121
Start Time (UTC)
2024-12-12T12:00:00
End Time (UTC)
2024-10-12T14:00:00
Add Shift
Update Shift List
Shift List
Shift ID
Start Time (UTC)
End Time (UTC)
0
0001-01-01T00:00:00Z
0001-01-01T00:00:00Z
1
0001-01-01T00:00:00Z
0001-01-01T00:00:00Z
121
2024-11-12T00:00:00Z
2024-11-12T00:00:00Z
232
0001-01-01T00:00:00Z
0001-01-01T00:00:00Z


SchedulerWebApp
localhost:5131/Allocate-Tasks
Scheduler
About
Task Allocation Form
Task ID
1
Task Name
meeting
Assign Staff
ali
Add Task Allocation
Task Allocations
Task ID
Task Name
Assigned Staff
11
meeting
ali
99
input data
Not Assigned


SchedulerWebApp
localhost:5131/Assign-Staffs
Scheduler
About
Add Shift Assignment
Staff Name
ali
Shift ID
1
Start Time (UTC)
2024-10-12T09:00:00
End Time (UTC)
2024-10-12T14:00:00
Add Shift Assignment
Refresh Shift List
No shift assignments added yet.

3.0 UML DIAGRAM



4.0 EVALUATION OF PLATFORM

1. Overview of Blazor

Blazor is a framework for building interactive web applications using C# and .NET instead of JavaScript. It comes in two main hosting models:

- **Blazor WebAssembly (WASM):** Runs in the browser via WebAssembly.
- **Blazor Server:** Runs on the server and communicates with the client over SignalR.

2. Strengths of Blazor

a. Full-Stack C# Development

Enables developers to use C# for both client-side and server-side code, reducing the need for JavaScript.

Allows sharing of libraries, models, and business logic across the application.

b. Integration with .NET Ecosystem

Seamless integration with existing .NET libraries, APIs, and tooling like Entity Framework Core. Leverages Visual Studio for rich development experience (debugging, IntelliSense).

3. Weaknesses of Blazor

a. Performance Limitations

Blazor WebAssembly: Initial load time can be high due to downloading the runtime and dependencies.

Blazor Server: Relies on SignalR, which may introduce latency for applications with high interaction rates or poor network conditions.

b. Browser Support

Requires modern browsers supporting WebAssembly or JavaScript (for interop).

6. Comparison with Other Frameworks

Feature	Blazor WebAssembly	Blazor Server
Language	C#	C#
Performance	Slower initial load	Real-time but relies on SignalR
Hosting Model	Client-Side	Server-Side
Ecosystem	Growing	Growing
Ease of Use	High for .NET devs	High for .NET devs

7. Conclusion

Blazor is a robust platform for developers familiar with C# and .NET, offering a modern approach to building web applications. It's particularly well-suited for scenarios requiring tight integration with .NET backends. However, its smaller ecosystem and potential performance constraints in Blazor WebAssembly may require careful consideration for large-scale consumer-facing applications.