

Assignment 1: FinTech Dashboard (Blazor WASM + API + Charts)

Domain: FinTech

Time: ~60 minutes

Scenario

You are building a **personal finance dashboard** that shows monthly expenses using charts.

Objectives

- Consume REST API from Blazor WASM
- Render charts
- Change chart type at runtime

Tasks

1. Create a **Blazor WebAssembly** app (.NET 8/9/10)
2. Create a **mock API** (or in-memory service) returning:

[

```
{ "month": "Jan", "amount": 12000 },  
{ "month": "Feb", "amount": 9000 },  
{ "month": "Mar", "amount": 15000 }
```

]

3. Display data using a chart (Line or Bar)
4. Add a **Dropdown** to switch chart type at runtime:
 - Bar
 - Line
 - Pie
5. Restrict chart size to **200x200**

Expected Output

- Chart renders correctly
- Chart type switches without page reload

Concepts Reinforced

- HttpClient in Blazor
 - Component state
 - Dynamic UI rendering
-

Assignment 2: Real-Time Chat Support (Blazor Server + SignalR)

Domain: Customer Support / SaaS

Time: ~60 minutes

Scenario

Create a **live customer support chat** system.

Objectives

- Understand SignalR
- Real-time communication
- Hub ↔ Client messaging

Tasks

1. Create a **Blazor Server** app

2. Add a **SignalR Hub**:

Task SendMessage(string user, string message)

3. Create a Razor component:

- Input for username
- Input for message
- Chat message list

4. Broadcast messages to all connected users

5. Highlight:

- Sender name
- Timestamp

Expected Output

- Multiple browser tabs can chat in real time

Concepts Reinforced

- SignalR Hub
 - Persistent connections
 - Real-time UI updates
-

Assignment 3: Secure Admin Panel (Blazor Server + Identity)

Domain: Enterprise / Internal Tools

Time: ~60 minutes

Scenario

Build a **secure admin panel** accessible only to Admin users.

Objectives

- Authentication & Authorization
- Role-based access
- Protected routes

Tasks

1. Create a **Blazor Server** app

2. Configure **ASP.NET Core Identity**

3. Create roles:

- Admin
- User

4. Seed one Admin user

5. Create:

- `/login`
- `/admin-dashboard`

6. Protect /admin-dashboard:

- Only Admin role allowed

Expected Output

- Unauthorized users redirected
- Admin sees dashboard

Concepts Reinforced

- Identity
- Authorization policies
- Secure routing