ProfileBook – Social Media Web Application

Final Capstone Project Report

Submitted by:
Shaik.Vaseem

Batch: Wipro NGA- .Net Full Stack Angular- FY26 -C2

Instructor: Mrs. Jyothi S. Patil

Date: September 2025

CERTIFICATE

This certificate is awarded to **Shaik.Vaseem** for the successful completion of the project titled **"ProfileBook – Social Media Web Application."** This project was undertaken as a requirement for the Pre-Skill Training program jointly conducted by **Wipro and Great Learning**.This report is a sincere and original record of the work accomplished by the student. The project documentation and its contents have not been previously submitted to any other institution for any academic award.

Project Coordinator

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported me throughout the course of this project, **"ProfileBook – Social Media Web Application”**

My deepest thanks go to my project coordinator, **Ms. Nissy Joy**, for her invaluable guidance, unwavering support, and constructive feedback. Her insights were instrumental in shaping the direction of this project.

I am also incredibly thankful to my mentor, **Mrs. Jyothi S. Patil**, for her patient mentorship, encouragement, and expert advice. Her deep knowledge and practical guidance were crucial in overcoming challenges and refining the project's outcome.

DECLARATION

I, **Shaik.Vaseem**, do solemnly affirm that the project report titled **"ProfileBook – Social Media Web Application"** is my original work and has been undertaken to fulfill the requirements for the Pre-Skill Training assigned by **Wipro and Great Learning**. I confirm that this report, in whole or in part, has not been submitted for any other academic purpose or to any other institution.

Table of Contents

Problem Definition and Objectives

1.1 Problem Definition

In today's digital-first society, online social platforms are an integral part of human interaction. However, many existing platforms are either:

- **Overly complex** for new users.
- **Lacking moderation** controls, leading to harmful or inappropriate content.
- **Weak in security**, risking privacy breaches.

The **ProfileBook** project aims to build a **scalable, secure, and user-friendly social media platform** that balances **user freedom** with **administrative oversight**.

1.2 Objectives

The main objectives of ProfileBook are:

1. **User-Centric Interaction:** Allow users to register, create posts, like/comment, and connect with others.
2. **Secure Authentication:** Implement JWT token-based login to ensure user data confidentiality.
3. **Content Moderation:** Allow admins to approve/reject posts and handle reports.
4. **Efficient Communication:** Enable private messaging and (optional) real-time chat.
5. **Database Reliability:** Store and manage posts, users, and reports effectively in SQL Server.
6. **Scalability:** Architect the system so that it can be deployed on cloud infrastructure in the future

Technology Stack

2.1 Frontend – Angular

- **Framework:** Angular 17
- **Key Features:**
    - Component-driven development.
    - Angular Router for navigation.
    - FormsModule and ReactiveFormsModule for handling forms.
    - State management (NgRx/Redux) for login persistence.

2.2 Backend – ASP.NET Core Web API

- **Framework:** ASP.NET Core 9.0
- **Design Pattern:** MVC (Model-View-Controller) with Web API endpoints.
- **Security:** JWT authentication and authorization middleware.
- **ORM:** Entity Framework Core for CRUD operations.

2.3 Database – SQL Server

- **Entities:** Users, Posts, Messages, Reports, Groups, Likes, Comments.
- **ORM Mapping:** Entity Framework Core automatically handles schema updates.
- **Indexes & Optimizations:** Indexed frequently queried columns (e.g., `UserId`, `PostId`).

2.4 Tools and Libraries

- **Swagger:** API documentation.
- **SignalR:** Real-time communication (optional).

System Architecture Overview

3.1 High-Level Flow

1. User opens Angular frontend.
2. Angular sends HTTP requests to ASP.NET Core Web API.
3. Web API validates request, processes logic, interacts with SQL Server.
4. Data is returned as JSON to Angular frontend.

3.2 Security Workflow

- JWT token is issued upon successful login.
- Token is sent with every API request.
- Backend validates token and user role (user/admin).

Angular (Frontend) → Web API (Backend) → SQL Server (Database).

User and Admin Stories

4.1 User Stories

- **Registration & Login:** Secure sign-up and login.
- **Create Posts:** Submit posts with optional images.
- **View Posts:** View only approved posts from admin.
- **Like/Comment:** Engage with posts through likes and comments.
- **Messaging:** Private communication between users.
- **Report Users:** Flag inappropriate behavior.
- **Search Users:** Search based on username/profile.

4.2 Admin Stories

- **Authentication:** Admin login with elevated privileges.
- **User Management:** CRUD operations on users.
- **Post Approval:** Accept/reject pending posts.
- **Report Handling:** View and act on reported users.
- **Group Management:** Create user groups for targeted communication.

Frontend Design (Angular)

5.1 Components

- **Auth Components:** `LoginComponent`, `RegisterComponent`.
- **Post Components:** `PostListComponent`, `PostCreateComponent`, `PostDetailComponent`.
- **Messaging:** `MessageListComponent`, `MessageSendComponent`.
- **Admin:** `AdminDashboardComponent`, `UserManagementComponent`, `ReportManagementComponent`.

5.2 Routing

- `/login` – Login page.
- `/register` – Registration page.
- `/posts` – Post listing.
- `/messages` – User messaging.
- `/admin` – Admin dashboard.

Backend Design (ASP.NET Core Web API)

6.1 Controllers

- **UserController:** Register, login, fetch profile.
- **PostController:** Create, approve, fetch posts.
- **MessageController:** Send, fetch messages.
- **ReportController:** Submit/view reports.

6.2 Security

- Middleware checks JWT token on each request.
- Role-based access: *User* vs *Admin*.

6.3 Business Layer

- Services layer validates input and applies business logic.
- Example: A post is only visible if its `status = Approved`.

Database Design (Schema + ERD)

7.1 Tables

1. **Users:** `UserId`, `Username`, `PasswordHash`, `Role`, `ProfileImage`.
2. **Posts:** `PostId`, `UserId`, `Content`, `PostImage`, `Status`.
3. **Messages:** `MessageId`, `SenderId`, `ReceiverId`, `Content`, `Timestamp`.
4. **Reports:** `ReportId`, `ReportedUserId`, `Reason`, `Timestamp`.
5. **Groups:** `GroupId`, `GroupName`, `Members`.
6. **Likes:** `LikeId`, `PostId`, `UserId`.
7. **Comments:** `CommentId`, `PostId`, `UserId`, `Content`, `Timestamp`.

7.2 Relationships

- One-to-Many between Users → Posts.
- One-to-Many between Users → Messages.
- One-to-Many between Posts → Comments.
- Many-to-Many between Users → Groups.

Component Breakdown & API Endpoints

8.1 Sample Endpoints

- `POST /api/auth/login` → Authenticate user.
- `POST /api/auth/register` → Register new user.
- `GET /api/posts` → Fetch approved posts.
- `POST /api/posts` → Create new post.
- `PUT /api/posts/{id}/approve` → Approve post (Admin).
- `POST /api/messages` → Send message.
- `POST /api/reports` → Report user.

8.2 API Documentation

- Swagger UI integrated for testing endpoints.

Sprint Implementation Plan

Sprint I (Planning & Setup)

- Use Case Documentation.
- Database Schema & ERD.
- Backend skeleton with placeholder controllers.
- Angular project initialization with static pages.

Sprint II (Core Development)

- Implement user authentication with JWT.
- CRUD operations for posts and messages.
- Angular integration with API.
- Admin panel design.

Sprint III (Enhancements & Finalization)

- Search & Filter functionality.
- Post approval workflow.
- Report handling by admin.
- Swagger API documentation.
- End-to-end testing.

10. Deployment Strategy

- **Version Control:** GitHub private repo.
- **Local Deployment:** IIS/Visual Studio for backend, Angular CLI for frontend.
- **Future Cloud Deployment:** Azure App Service (Backend), Azure SQL Database (DB).
- **Continuous Integration:** GitHub Actions for automated builds.

---

11. Testing & Quality Assurance

Testing Types

- **Unit Testing:** xUnit (Backend), Karma (Frontend).
- **Integration Testing:** Postman, Swagger.
- **Performance Testing:** JMeter (Optional).
- **UAT:** Manual testing by peers and mentor.

Sample Test Cases

- Valid login returns JWT.
- Invalid login returns Unauthorized (401).
- Admin approval changes post status.
- Reported users are visible to Admin only.

---

12. Challenges Faced & Solutions

- **State Persistence:** Solved using Redux store in Angular.
- **File Upload Handling:** Used `IFormFile` in ASP.NET.
- **Database Migrations:** EF Core migrations for schema updates.
- **Authentication Errors:** Debugged JWT middleware setup.

---

13. Future Enhancements

- Cloud deployment with CI/CD.
- Real-time chat with SignalR.
- AI-based content moderation.
- Mobile app version.
- Role-based dashboards for moderators.

---

14. Conclusion

ProfileBook provides a **functional and secure social media platform** with both user features and admin moderation. It demonstrates full-stack development using **Angular + ASP.NET Core + SQL Server**, aligning with modern enterprise applications.

This project is a **capstone achievement** in my learning journey under **Wipro NGA-Net Full Stack Angular FY26-C2** and guided by my mentor **Joythi S Patil.**