# Project ID- P5;

**Project Name -** Expense Splitter App

# Project Description:

A tool that helps users split expenses among friends and track payments with a real- time database.

# Key Concepts Covered:

React, Redux, Context API, JWT Authentication, REST API, Bootstrap

# Capstone project directory structure

/FullStackApp

│── /client (React Frontend)

│ ├── /public

│ ├── /src

│ │ ├── /components

│ │ ├── /pages

│ │ ├── /store (Redux Store)

│ │ ├── /services (API Calls)

│ │ ├── App.js

│ │ ├── index.js

│ └── package.json

│

│── /server (ASP.NET Core API)

│ ├── /Controllers

│ ├── /Models

│ ├── /Data (Database Context)

│ ├── /Services

│ ├── /Middleware

│ ├── appsettings.json

│ ├── Startup.cs

│ ├── Program.cs

│ └── FullStackApp.sln

│

│── /database

│ ├── setup.sql (DB Schema and Seed Data)

│ ├── migrations/

│ ├── README.md

│

│── README.md

│── .gitignore

│── docker-compose.yml (Optional)

|  |  |  |  |
| --- | --- | --- | --- |
| **Project** | **User Story** | **Description** | **Acceptance Criteria** |

|  |  |  |  |
| --- | --- | --- | --- |
| **P5 - Expense Splitter App** | Create a Group | Create a group to split expenses. | Store groups in MS SQL, allow invitations. |
| Add an Expense | Log expenses per group. | Display who owes what. |
| Settle Payments | Mark expenses as settled. | Update balances accordingly. |

Evaluation Rubric for Project Assessment (85 Marks)

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Excellent (Full Marks)** | **Needs Improvement (Partial Marks)** | **Poor (No Marks)** |
| **React Frontend (60 Marks)**  **(Each criterion is 10 marks, 6 sub-criteria)** | | | |
| UI/UX Design | Clean, responsive, and user- friendly interface | Some UI issues, minor responsiveness problems | Poor UI, broken responsiveness |
| Component Reusability | Proper use of modular and reusable components | Some redundant components, minimal reusability | No reusable components, messy code |
| State Management | Efficient use of React hooks, Context API, or Redux | Some issues in state handling, unnecessary re- renders | Poor state management, frequent bugs |
| API Integration | API calls handled efficiently with error handling | Basic API calls but lacks proper error handling | API calls fail, no proper state update |

|  |  |  |  |
| --- | --- | --- | --- |
| Routing & Navigation | Proper use of React Router, seamless navigation | Some broken links or inefficient routing | No routing implemented or broken links |
| Performance Optimization | Uses memoization, lazy loading, and proper key handling | Some performance lags, missing optimizations | Poor performance, slow load times |
| **.NET Core Web API (15 Marks) (Each criterion is 5 marks, 3 sub-criteria)** | | | |
| API Design & Structure | RESTful API with proper naming conventions, modular structure | Some inconsistent API routes, minor structural issues | Poorly designed API, inconsistent endpoints |
| Business Logic & Validation | Properly implemented business logic with input validation | Some missing validation checks, minor logical flaws | Major logical errors, missing validation |
| Security & Error Handling | Implements authentication, authorization, and error handling | Some security flaws, minimal error handling | No security, major vulnerabilities |
| **Database (10 Marks)**  **(Each criterion is 5 marks, 2 sub-criteria)** | | | |
| Schema Design | Well-structured, normalized database schema | Some redundant data, minor normalization issues | Poor schema design, redundant data |
| Query Performance | Optimized queries, proper indexing | Some unoptimized queries, minor performance issues | Poor query optimization, slow execution |

VIVA Evaluation Rubric (15 Marks)

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | Excellent (Full Marks) | Needs Improvement (Partial Marks) | Poor (No Marks) |
| **Documentation, Viva, Presentation, Report (15Marks)** | | | |
| Project Report & Documentation | Well-written documentation with clear API references | Incomplete documentation, missing some details | No documentation, poorly written |
| Viva & Presentation | Clear understanding of project, good presentation skills | Some difficulty explaining concepts | Poor explanation, lacks clarity |