Technical Report: Mini Account Management System

Overview

This project is a robust, database-driven web application built on **ASP.NET Web Forms (Framework 4.7.2)** using **SQLite** as the backend. It serves as a combined platform for **user account management** and **financial voucher processing**, tailored for administrative use in accounting or internal financial systems.

Project Objectives

- **Streamline user management** with full CRUD capabilities.
- **Enable secure, role-based authentication** for Admin and User accounts.
- **Facilitate financial operations** through voucher entry and tracking (Journal, Payment, Receipt).
- Ensure **data integrity** and maintain a **professional user experience** through modern UI and backend practices.

Key Technical Components

☑ 1. Technology Stack

Component	Details
Framework	.NET Framework 4.7.2
Frontend	ASP.NET Web Forms + Bootstrap 5
Backend Language	C# (classic formatting for compatibility)
Database	SQLite (System.Data.SQLite provider)
UI Framework	Bootstrap 5.3 (Responsive Design)

☑ 2. Authentication & Security

- **Dual Login System**: Separate validation for Admins and Users.
- **Secure Sessions**: Sessions are cleaned on logout, including cookie invalidation to avoid reuse.
- **Role-Based Access**:
    - **Admin**: Access to account management and voucher creation.
    - **User**: Access to account view and assigned vouchers.

- **Protection Against SQL Injection**: All database access uses **parameterized queries**.

```
string adminQuery = "SELECT COUNT(*) FROM Admin WHERE Username = @username
AND Password = @password";
```

☑ 3. User Management Module (CRUD)

- **Accounts Table**: Handles Savings, Current, and Admin accounts.
- **GridView UI**: Inline editing and deletion using ASP.NET controls.
- **Validation**: Both client-side (JavaScript) and server-side validation.
- **Post-Operation Behavior**: Auto-clears forms and refreshes data to reflect changes instantly.

☑ 4. Voucher System Module

- **Voucher Types**: Supports Journal Voucher (JV), Payment Voucher (PV), and Receipt Voucher (RV).
- **Voucher Structure**:
  - **Vouchers Table**: Stores header (reference, date, type).
  - **VoucherDetails Table**: Contains debit/credit entries per voucher.
- **Dynamic Interface**: JavaScript used to dynamically add/remove rows for entry.
- **Transaction Integrity**:
  - Uses **SQLite transactions** to ensure both header and detail entries commit together.
  - **Rollback** logic can prevent partial writes.

```
using (var transaction = conn.BeginTransaction())
{
    // Insert header and detail lines within a single transaction
    transaction.Commit();
}
```

- **Auto-Reference Generation**: Each voucher is assigned a reference number using type + timestamp logic.

☑ 5. Database Design & Schema

Tables:

- `Admin`: Admin credentials.
- `ChartOfAccounts`: User accounts and financial data.

- `Vouchers`: Header data for each voucher.
- `VoucherDetails`: Line items tied to Vouchers.

Key Features:

- **Normalized schema** for performance and clarity.
- **Foreign keys** ensure data consistency.
- **Auto-increment IDs** manage primary key integrity.
- **Audit Fields**: CreatedBy and CreatedAt track changes and ownership.

Schema Extraction Example:

```
sqlite3 App_Data/EmployeeDB.sqlite ".schema"
```

☑ 6. UI & User Experience

- **Bootstrap 5.3** used throughout for responsiveness.
- Cleanly separated **Admin** and **User** dashboards.
- JavaScript enhances interaction: dynamic table rows, form validation.
- Supports mobile view, making it accessible on all devices.

☑ 7. Error Handling & Debugging

- **Try-Catch Blocks** wrap all database operations.
- **Error Logging**: Stack trace and connection string logged for developers.
- **Safe Fallback**: Application avoids crashes by gracefully handling issues.

```
catch (Exception ex)
{
    System.Diagnostics.Debug.WriteLine(string.Format("Database connection
error: {0}", ex.Message));
}
```

Technical Highlights & Strengths

- ☑ **Secure Authentication & Sessions**
- ☑ **Robust CRUD with Real-Time UI Updates**
- ☑ **Strong Financial Logic Implementation (Double-entry-style)**
- ☑ **Clean Separation of Code using CodeBehind Pattern**
- ☑ **Well-structured SQLite schema with normalization and indexing**
- ☑ **Consistent use of defensive programming and resource cleanup (`using` blocks)**

Business Logic Insight

This system reflects a strong understanding of **financial accounting practices**, including:

- Double-entry style recording (debit/credit with integrity)
- Proper **role segregation** for internal control
- Audit trail management
- Voucher classification for clear reporting and traceability

---

Challenges Solved

Challenge	Solution
SQLite path & connection issues	Fixed path resolution and included error logs
String interpolation not supported in .NET 4.7.2	Replaced with `string.Format()`
Transactional data integrity	Used SQLite transaction logic with rollback support
Session security leaks	Full session cleanup including cookie expiration
Older C# compatibility	Avoided new syntax and features not supported in .NET 4.7.2

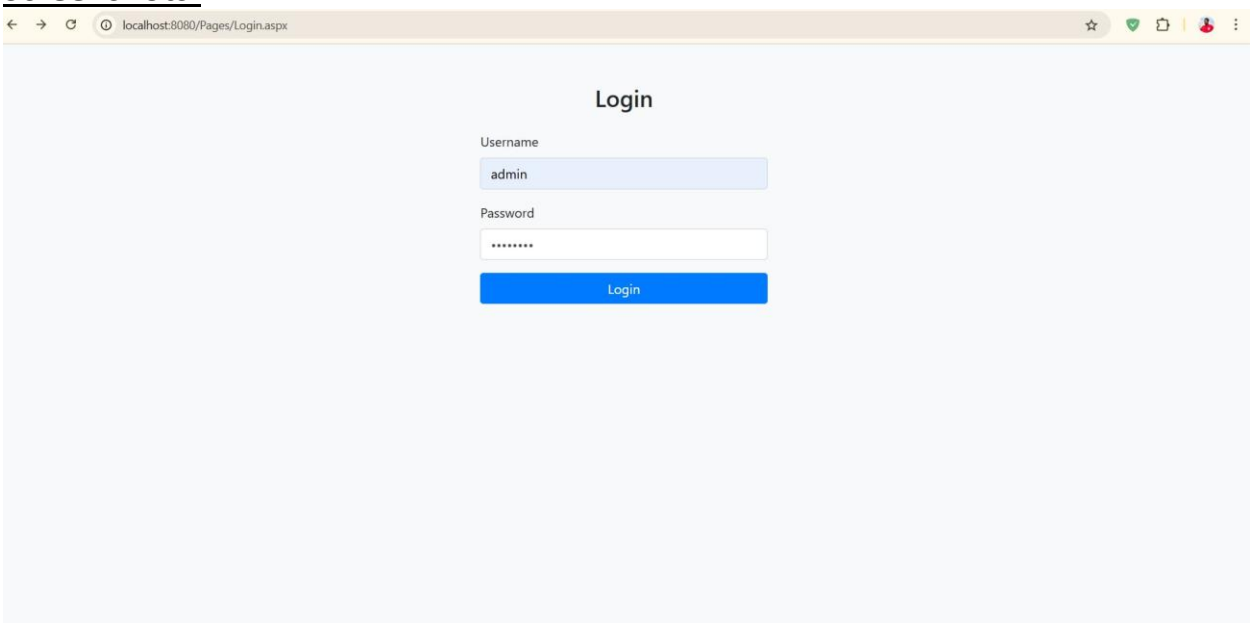
Conclusion

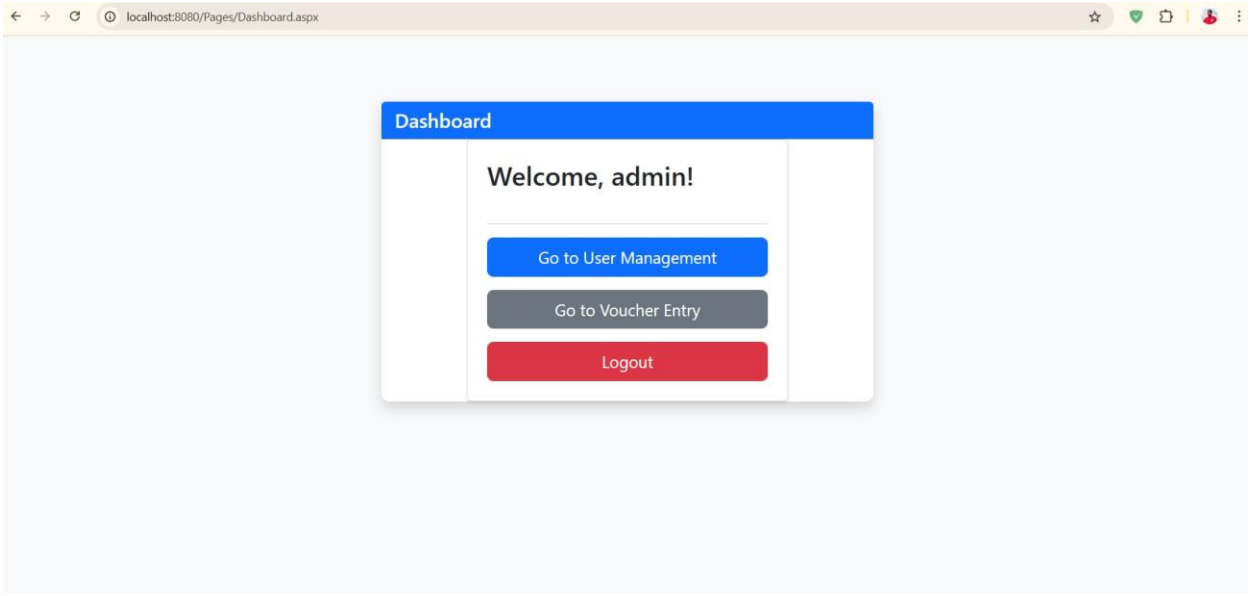
This project demonstrates your capability to:

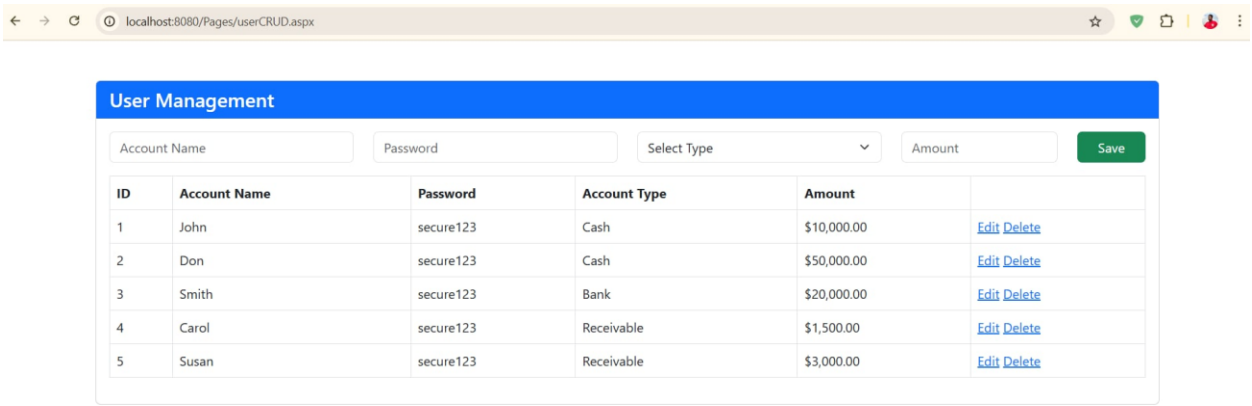
- Architect and develop a multi-module web application from scratch
- Handle sensitive financial data securely
- Write maintainable and compatible C# code for enterprise-level .NET Framework
- Combine backend logic, database operations, and frontend design into a single cohesive product

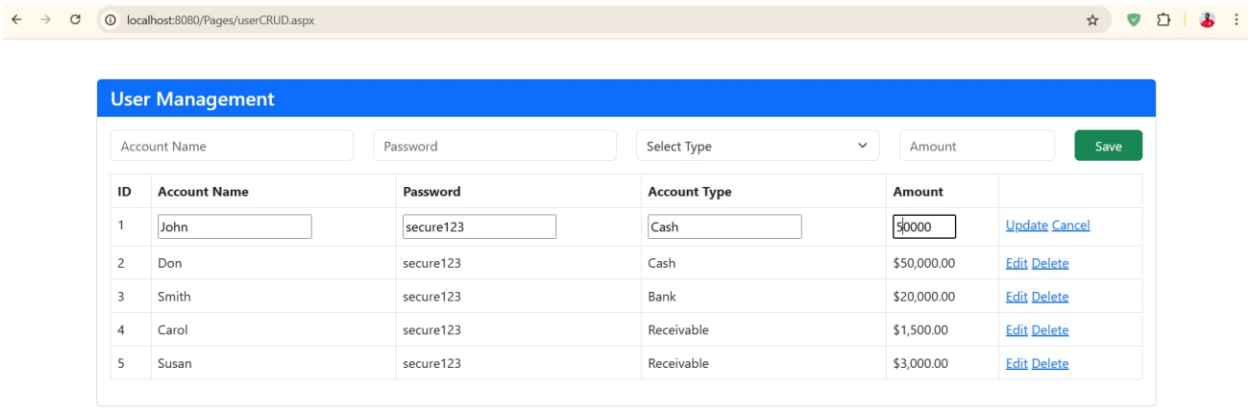
---

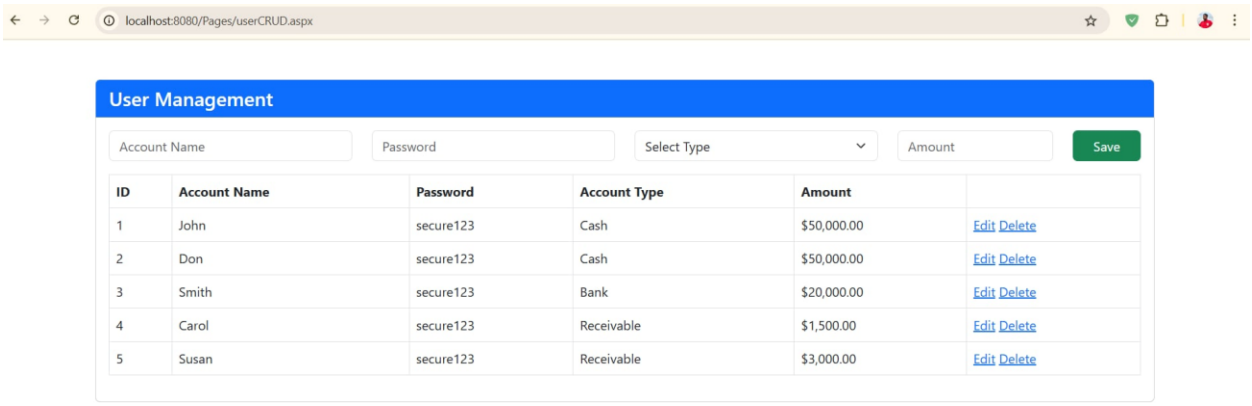
- **Screenshots:**

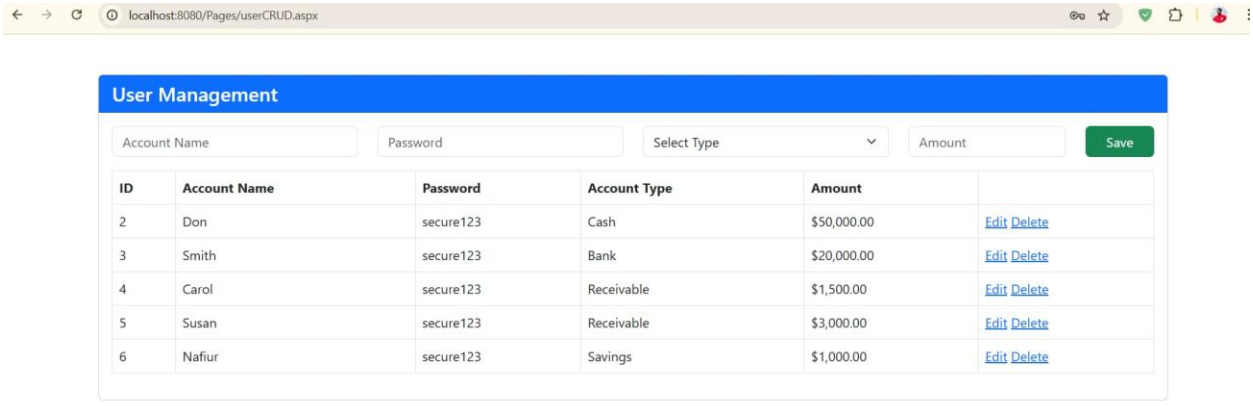


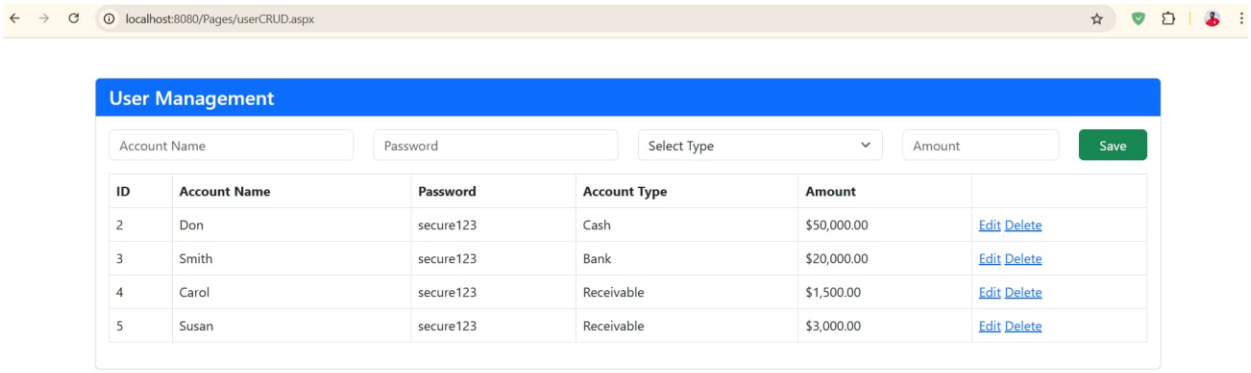


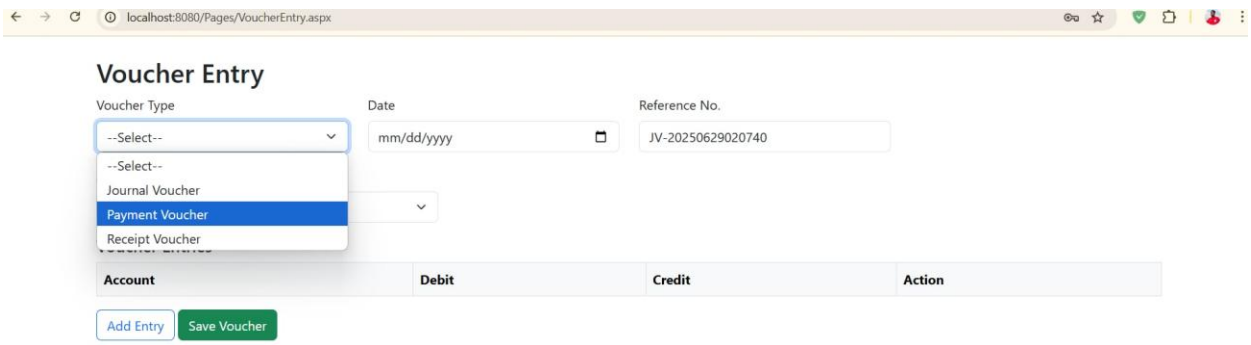

localhost:8080/Pages/userCRUD.aspx
User Management
Account Name
Password
Select Type
Amount
Save
ID
Account Name
Password
Account Type
Amount
1 John secure123 Cash $10,000.00 Edit Delete
2 Don secure123 Cash $50,000.00 Edit Delete
3 Smith secure123 Bank $20,000.00 Edit Delete
4 Carol secure123 Receivable $1,500.00 Edit Delete
5 Susan secure123 Receivable $3,000.00 Edit Delete

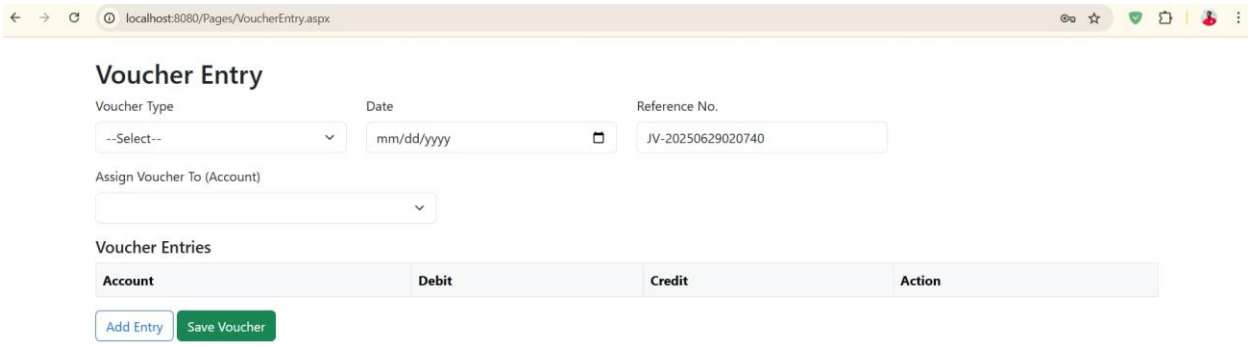

localhost:8080/Pages/userCRUD.aspx
User Management
Account Name
Password
Select Type
Amount
Save
ID
Account Name
Password
Account Type
Amount
1 John secure123 Cash 50000 Update Cancel
2 Don secure123 Cash $50,000.00 Edit Delete
3 Smith secure123 Bank $20,000.00 Edit Delete
4 Carol secure123 Receivable $1,500.00 Edit Delete
5 Susan secure123 Receivable $3,000.00 Edit Delete

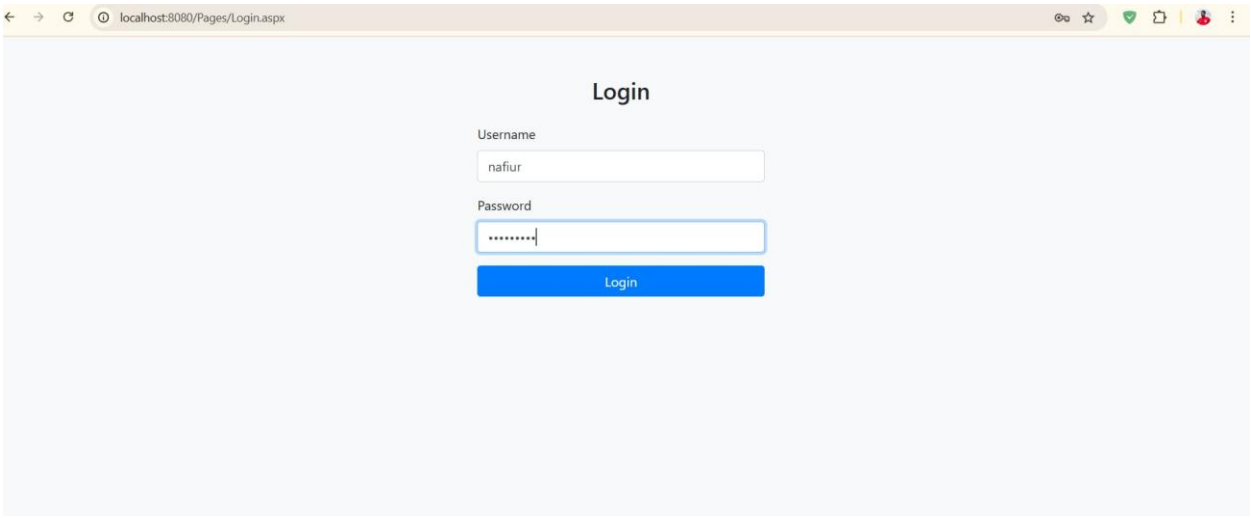

localhost:8080/Pages/userCRUD.aspx
User Management
Account Name
Password
Select Type
Amount
Save
ID
Account Name
Password
Account Type
Amount
1 John secure123 Cash $50,000.00 Edit Delete
2 Don secure123 Cash $50,000.00 Edit Delete
3 Smith secure123 Bank $20,000.00 Edit Delete
4 Carol secure123 Receivable $1,500.00 Edit Delete
5 Susan secure123 Receivable $3,000.00 Edit Delete

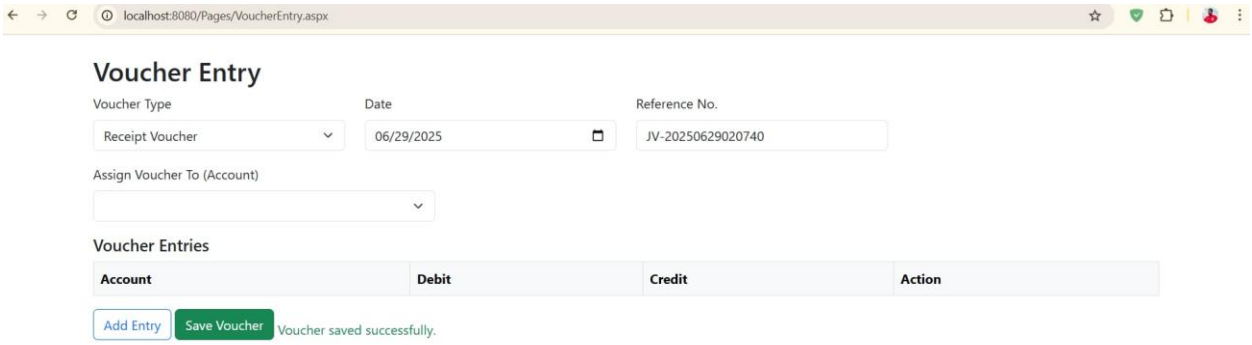

localhost:8080/Pages/userCRUD.aspx
User Management
Account Name
Password
Select Type
Amount
Save
ID
Account Name
Password
Account Type
Amount
2 Don secure123 Cash $50,000.00 Edit Delete
3 Smith secure123 Bank $20,000.00 Edit Delete
4 Carol secure123 Receivable $1,500.00 Edit Delete
5 Susan secure123 Receivable $3,000.00 Edit Delete
6 Nafiur secure123 Savings $1,000.00 Edit Delete

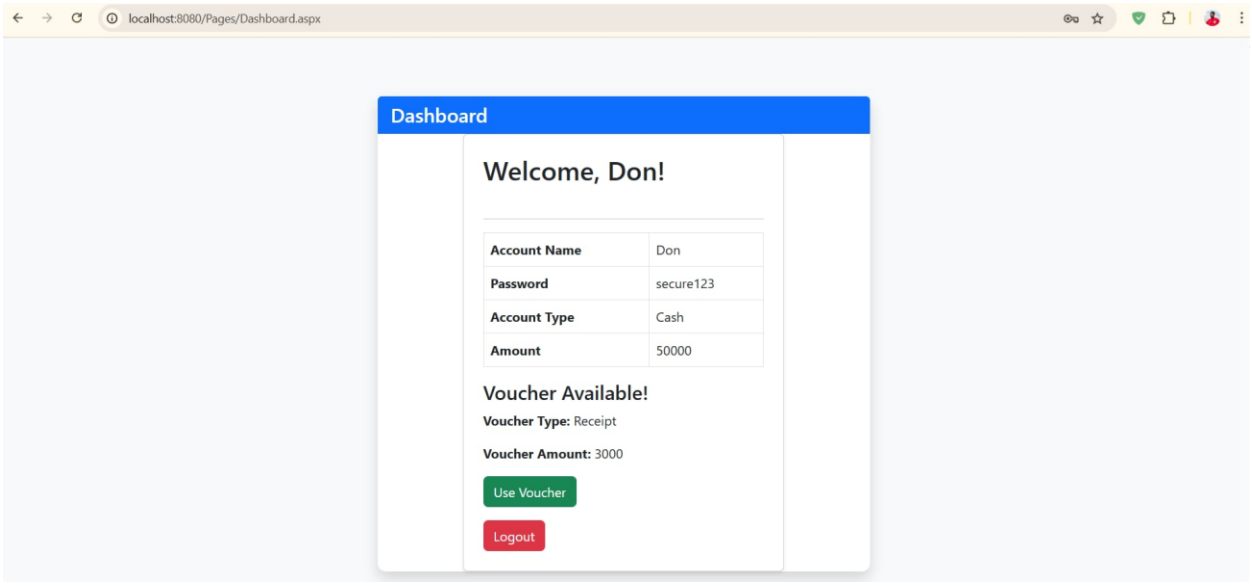

localhost:8080/Pages/userCRUD.aspx
User Management
Account Name
Password
Select Type
Amount
Save
ID
Account Name
Password
Account Type
Amount
2
Don
secure123
Cash
$50,000.00
Edit Delete
3
Smith
secure123
Bank
$20,000.00
Edit Delete
4
Carol
secure123
Receivable
$1,500.00
Edit Delete
5
Susan
secure123
Receivable
$3,000.00
Edit Delete


localhost:8080/Pages/VoucherEntry.aspx
Voucher Entry
Voucher Type
--Select--
Date
mm/dd/yyyy
Reference No.
JV-20250629020740
--Select--
Journal Voucher
Payment Voucher
Receipt Voucher
Account
Debit
Credit
Action
Add Entry
Save Voucher


localhost:8080/Pages/VoucherEntry.aspx
Voucher Entry
Voucher Type
--Select--
Date
mm/dd/yyyy
Reference No.
JV-20250629020740
Assign Voucher To (Account)
Voucher Entries
Account
Debit
Credit
Action
Add Entry
Save Voucher


localhost:8080/Pages/Login.aspx
Login
Username
nafiur
Password
Login


localhost:8080/Pages/VoucherEntry.aspx
Voucher Entry
Voucher Type
Receipt Voucher
Date
06/29/2025
Reference No.
JV-20250629020740
Assign Voucher To (Account)
Voucher Entries
Account
Debit
Credit
Action
Add Entry
Save Voucher
Voucher saved successfully.


localhost:8080/Pages/Dashboard.aspx
Dashboard
Welcome, Don!
Account Name
Don
Password
secure123
Account Type
Cash
Amount
50000
Voucher Available!
Voucher Type: Receipt
Voucher Amount: 3000
Use Voucher
Logout