




 Author

 Name: Md Mahabubul Alam Abir

 Email: mahbubabir09@gmail.com

 Role: Junior .NET Developer (Test Submission)

📄 Mini Account Management System

A basic accounting application built using ASP.NET Core Razor Pages, SQL Server, and stored procedures — developed for a Junior .NET Developer test task.

🚀 Technologies Used

- **ASP.NET Core Razor Pages (.NET 8)**
- **MS SQL Server**
- **Stored Procedures only (No LINQ)**
- **ASP.NET Identity (custom roles)**
- **Bootstrap 5**
- **ClosedXML (Excel export)**

📷 Screenshots

🗝️ Register and Login Page

The screenshot shows the 'Register' page of the MiniAccountSystem. The page has a header with the site name and navigation links. The main content area is titled 'Register' and includes a sub-header 'Create a new account.' and a link 'Use another service to register.' Below this, there are three input fields: 'Email', 'Password', and 'Confirm Password'. A blue 'Register' button is at the bottom. A small text block on the right side of the form states: 'There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).'

The screenshot shows the 'Log in' page of the MiniAccountSystem. The page has a header with the site name and navigation links. The main content area is titled 'Log in' and includes a sub-header 'Use a local account to log in.' and a link 'Use another service to log in.' Below this, there are two input fields: 'Email' (containing 'mahubabir09@gmail.com') and 'Password' (masked with dots). There is a 'Remember me?' checkbox and a blue 'Log in' button. At the bottom, there are links for 'Forgot your password?', 'Register as a new user', and 'Resend email confirmation'. A small text block on the right side of the form states: 'There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).'

📄 Voucher Entry Form

The screenshot shows a web browser window with the address bar displaying 'localhost:7139/Vouchers/Create'. The page title is 'MiniAccountSystem' and the user is logged in as 'Hello mahbubabir09@gmail.com'. The main heading is 'Create Voucher'. Below it, there are three input fields: 'Voucher Type' with the value 'Journal', 'Voucher Date' with the value '06/30/2025', and 'Reference No' with the value '1'. Below these fields is a table titled 'Entries'. The table has three columns: 'Account', 'Debit', and 'Credit'. There are two rows of data. The first row has 'Cash' in the 'Account' column, '15000' in the 'Debit' column, and an empty 'Credit' column. The second row has 'Cash' in the 'Account' column, an empty 'Debit' column, and '15000' in the 'Credit' column. At the bottom of the table, there are two buttons: 'Add Row' and 'Save Voucher'. The Windows taskbar at the bottom shows the date and time as '11:06 PM 6/30/2025' and the weather as '28°C Haze'.

MiniAccountSystem Home Privacy Hello mahbubabir09@gmail.com Logout

Create Voucher

Voucher Type
Journal

Voucher Date
06/30/2025

Reference No
1

Entries

Account	Debit	Credit
Cash	15000	
Cash		15000

Add Row Save Voucher

📊 Chart of Accounts with Hierarchy

The screenshot shows a web browser window with the address bar displaying 'localhost:7139/Accounts/Manage'. The page title is 'MiniAccountSystem' and the user is logged in as 'Hello mahbubabir09@gmail.com'. The main heading is 'Create Account'. Below it, there are three input fields: 'AccountName', 'AccountType', and 'ParentId'. At the bottom of the form, there are two buttons: 'Save' and 'Cancel'. The Windows taskbar at the bottom shows the date and time as '11:17 PM 6/30/2025' and the weather as '28°C Haze'.

MiniAccountSystem Home Privacy Hello mahbubabir09@gmail.com Logout

Create Account

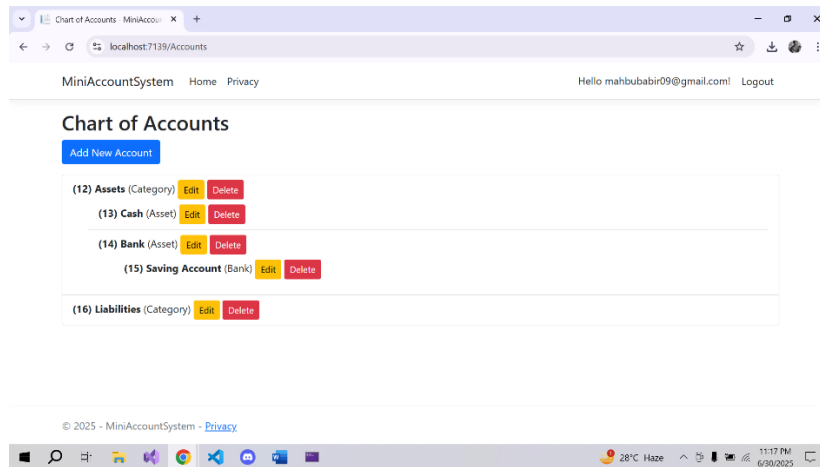
AccountName

AccountType

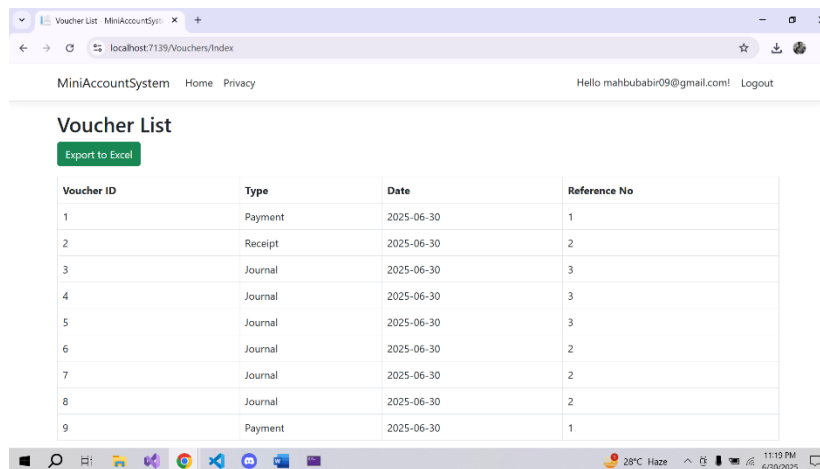
ParentId

Save Cancel

© 2025 - MiniAccountSystem - Privacy



📄 Voucher List with Export Option



📁 Core Features

✅ User Roles & Permissions

- Roles: `Admin`, `Accountant`, `Viewer`
- Role-based access to modules using ASP.NET Identity

Chart of Accounts

- Create/Edit/Delete accounts
- Parent/Child hierarchy supported
- Stored procedure: `sp_ManageChartOfAccounts`

Voucher Entry

- Types: Journal, Payment, Receipt
- Add multiple debit/credit lines
- Uses stored procedure: `sp_SaveVoucher`
- TVP used for bulk insert of entries

Voucher Listing

- View saved vouchers
- Filtered by date/type (can be added)
- Optional: detailed entries per voucher

Excel Export (Bonus)

- Export vouchers to `.xlsx` using `ClosedXML`
- Available on voucher list page

For Database Setup go to the next page

🗄 Database Setup

All database operations are handled using ****stored procedures**** and ****TVPs****.

****sp_ManageChartOfAccounts**

```
USE [MiniAccountDB]
```

```
GO
```

```
/****** Object: StoredProcedure [dbo].[sp_ManageChartOfAccounts]  Script Date: 6/30/2025 11:33:34 PM  
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
ALTER PROCEDURE [dbo].[sp_ManageChartOfAccounts]
```

```
    @Action NVARCHAR(10),    -- 'INSERT', 'UPDATE', 'DELETE'
```

```
    @AccountId INT = NULL,
```

```
    @AccountName NVARCHAR(100) = NULL,
```

```
    @ParentId INT = NULL,
```

```
    @AccountType NVARCHAR(50) = NULL
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    IF @Action = 'INSERT'
```

```
    BEGIN
```

```
        INSERT INTO ChartOfAccounts (AccountName, ParentId, AccountType)
```

```
        VALUES (@AccountName, @ParentId, @AccountType)
```

```
    END
```

```
    ELSE IF @Action = 'UPDATE'
```

```
    BEGIN
```

```

UPDATE ChartOfAccounts
SET AccountName = @AccountName,
    ParentId = @ParentId,
    AccountType = @AccountType
WHERE AccountId = @AccountId
END

ELSE IF @Action = 'DELETE'
BEGIN
    DELETE FROM ChartOfAccounts WHERE AccountId = @AccountId
END

END;

```

****TVP**

```

CREATE TYPE TVP_VoucherEntry AS TABLE
(
    AccountId INT,
    DebitAmount DECIMAL(18,2),
    CreditAmount DECIMAL(18,2)
);

```

****sp_SaveVoucher**

```

USE [MiniAccountDB]

GO

/***** Object: StoredProcedure [dbo].[sp_SaveVoucher]  Script Date: 6/30/2025 11:40:31 PM *****/

SET ANSI_NULLS ON

```

```

GO

SET QUOTED_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[sp_SaveVoucher]

    @VoucherType NVARCHAR(50),

    @VoucherDate DATE,

    @ReferenceNo NVARCHAR(100),

    @Entries TVP_VoucherEntry READONLY -- Table-valued parameter

AS

BEGIN

    SET NOCOUNT ON;


    INSERT INTO Vouchers (VoucherType, VoucherDate, ReferenceNo)

    VALUES (@VoucherType, @VoucherDate, @ReferenceNo);


    DECLARE @VoucherId INT = SCOPE_IDENTITY();


    INSERT INTO VoucherEntries (VoucherId, AccountId, DebitAmount, CreditAmount)

    SELECT @VoucherId, AccountId, DebitAmount, CreditAmount

    FROM @Entries;

END;

```

 Files:

- sp_ManageChartOfAccounts.sql: For insert/update/delete chart of accounts
- sp_SaveVoucher.sql: For saving voucher master and entries using TVP
- TVP_VoucherEntry.sql: Table-valued parameter type for entries

🖋️ How to Run the Project

1. ****Restore NuGet packages****:

dotnet restore

2. ****Configure DB connection string**** in `appsettings.json`:

```
``json
```

```
"ConnectionStrings": {
```

```
  "DefaultConnection":
```

```
  "Server=.;Database=MiniAccountDB;Trusted_Connection=True;TrustServerCertificate=True"
```

```
}
```

Apply Identity migration (only if needed):

dotnet ef database update

Run the app:

dotnet run

Open browser and go to:

<https://localhost:7139/>

📁 Folder Structure

/Pages

├— Accounts → Chart of Accounts (Create/Manage)

├— Vouchers → Voucher entry, listing, export

└— Shared → Layout and _LoginPartial