



Module Code & Module Title
CS6004NT- Application Development

Assessment Weightage & Type
Individual Coursework (30%)

Year and Semester
2024-25 Autumn

Student Name: Alish Ban

London Met ID: 22072045

College ID: NP05CP4A220011

Assignment Submission Date: December 19

Submitted To: Mr. Anant Singh

GitHub Link: <https://github.com/Ch3ra/SpendWise>

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded

Table of Contents

| | |
|------------------------------------|----|
| 1. Project Tittle: SpendWise | 1 |
| 2. Task 1 | 2 |
| 3. UI Design (Wireframe) | 8 |
| 3.1 Register | 9 |
| 3.2 Login..... | 10 |
| 3.3 Dashboard..... | 11 |
| 3.4 CashIn | 12 |
| 3.5 CashOut Flow..... | 13 |
| 4. Data/Entity Modeling..... | 14 |
| 4.1 Entity Relation Diagram..... | 16 |
| 5. Technology Stack | 17 |
| 6. References | 19 |

Table of Figures

| | |
|--|----|
| Figure 1 Creating new project | 2 |
| Figure 2 Selecting project templates | 3 |
| Figure 3 Setting project name SpendWise | 3 |
| Figure 4 Selecting framework..... | 4 |
| Figure 5 Project successfully created..... | 4 |
| Figure 6 Creating Repository | 5 |
| Figure 7 Signing in | 5 |
| Figure 8 Connecting visual studio folder with github | 6 |
| Figure 9 Connected successfully..... | 6 |
| Figure 10 Github profile after creation | 7 |
| Figure 11 Github repository | 7 |
| Figure 12 Figma | 8 |
| Figure 13 Register page of my project SpendWise | 9 |
| Figure 14 Login page of my project SpendWise..... | 10 |
| Figure 15 Dashboard of my project SpendWise | 11 |
| Figure 16 CashIn Flow page of my project SpenedWise..... | 12 |
| Figure 17 CashhOut Flow page of my project SpendWise..... | 13 |
| Figure 18 Entity name Users | 14 |
| Figure 19 Attributes name UserID | 15 |
| Figure 20 Relationship between the users and transaction | 15 |
| Figure 21 ERD of SpendWise | 16 |
| Figure 22 MAUI Blazor Hybrid Logo..... | 17 |

1. Project Tittle: SpendWise

SpendWise is a desktop application built using the .NET Framework and the C# programming language to simplify personal finance management. Designed for efficiency and ease of use, it enables users to easily track cash inflows, outflows, and debts through secure user authentication, intuitive data entry, and an interactive dashboard summarizing financial activities. Leveraging the robust features of C# and the .NET framework—renowned for strong typing, object-oriented programming capabilities, and seamless database integration SpendWise ensures reliable performance and accuracy. Developed using Visual Studio 2022, the application automates financial tracking, minimizes manual effort, and provides insightful summaries to help users make informed financial decisions, bridging the gap between traditional methods and modern digital solutions.

2. Task 1

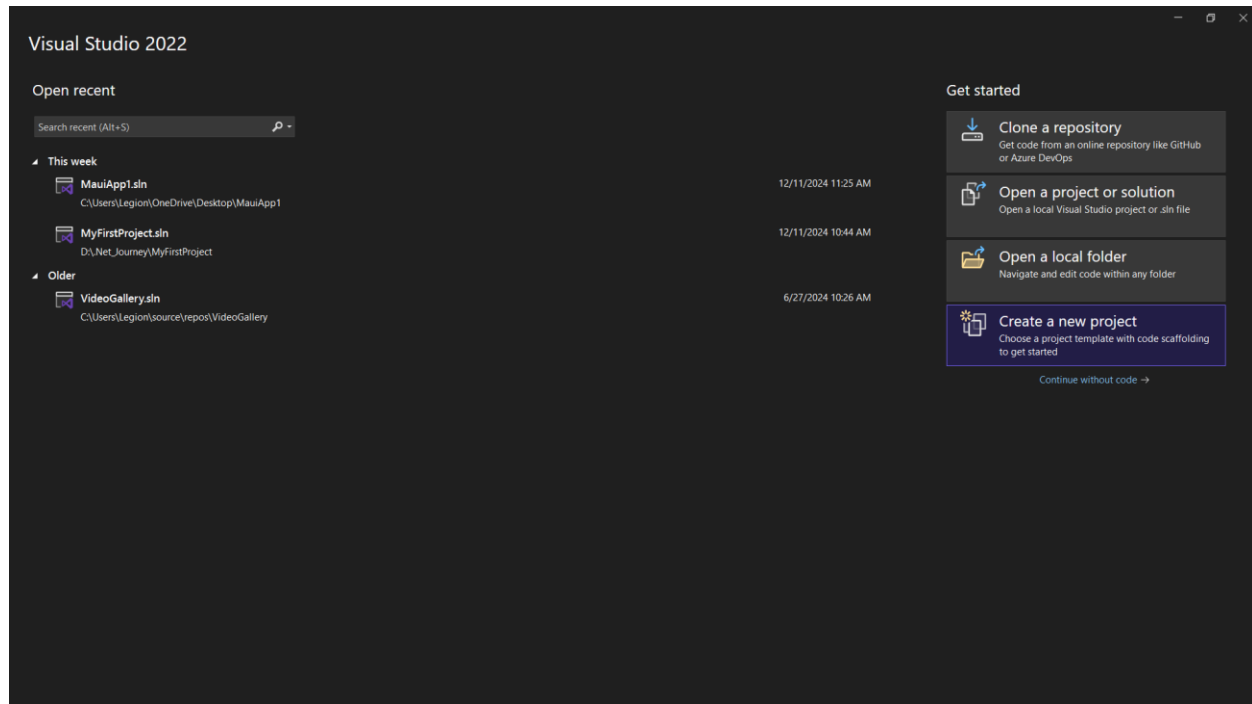


Figure 1 Creating new project

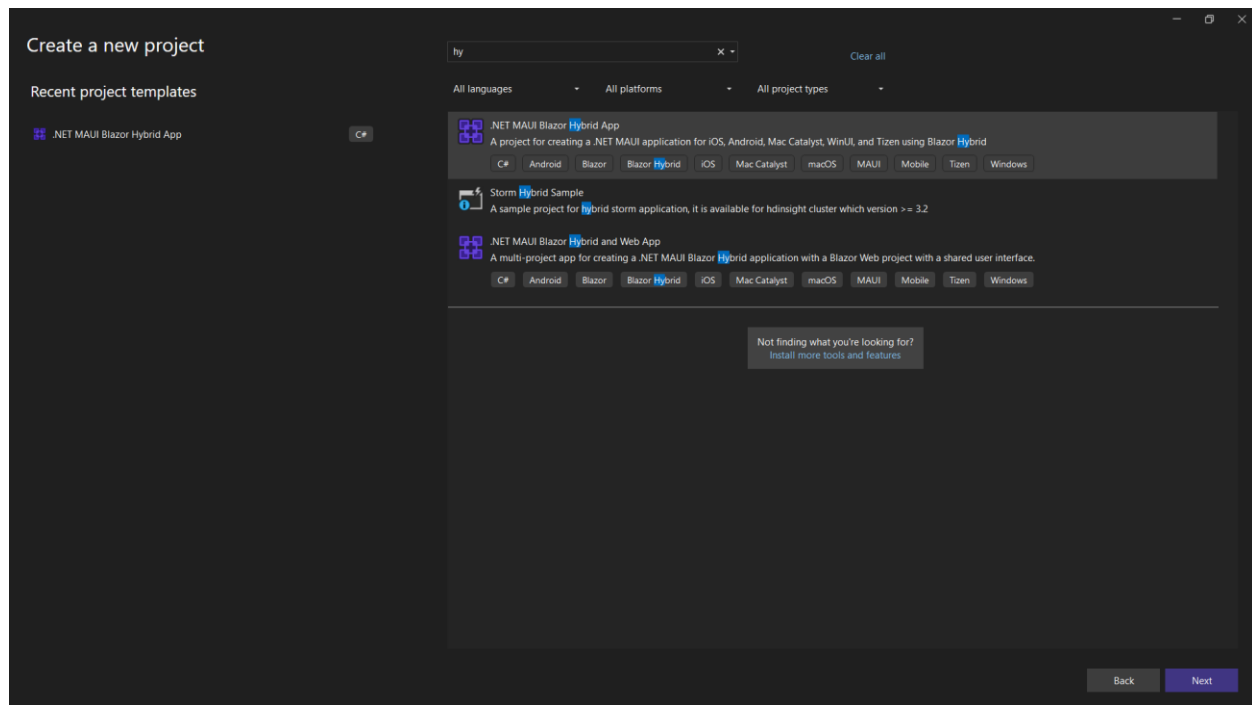


Figure 2 Selecting project templates

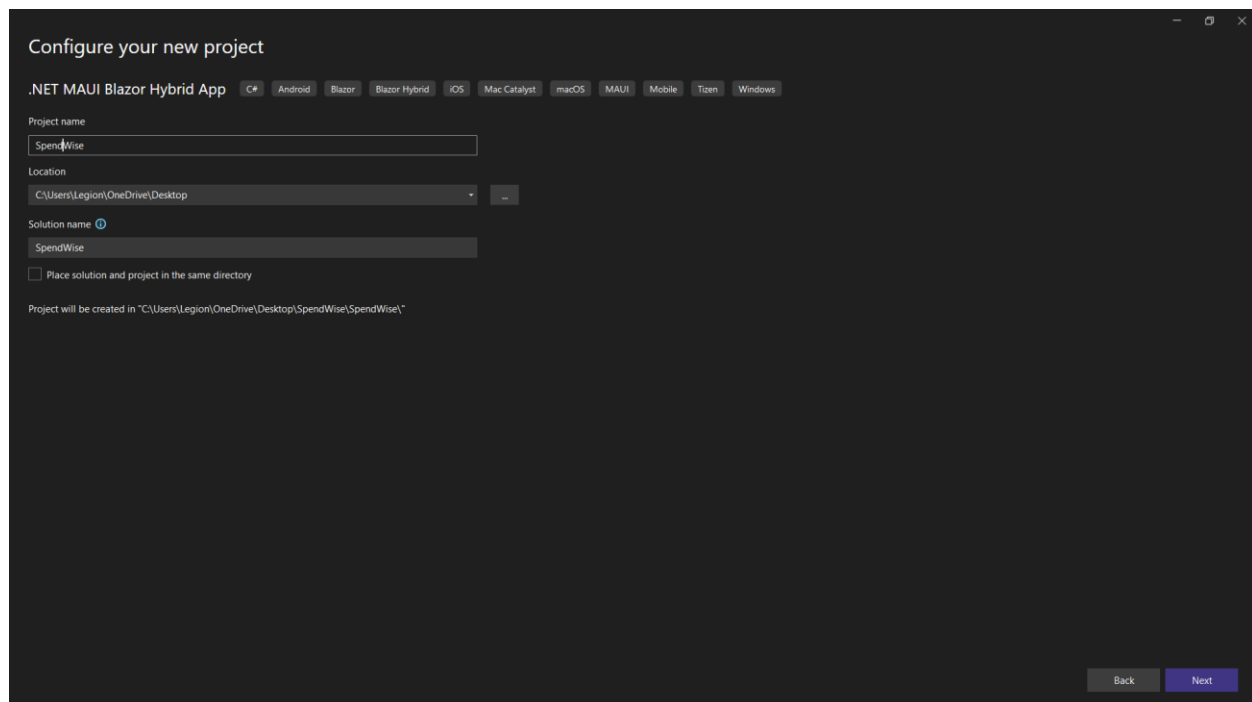


Figure 3 Setting project name SpendWise

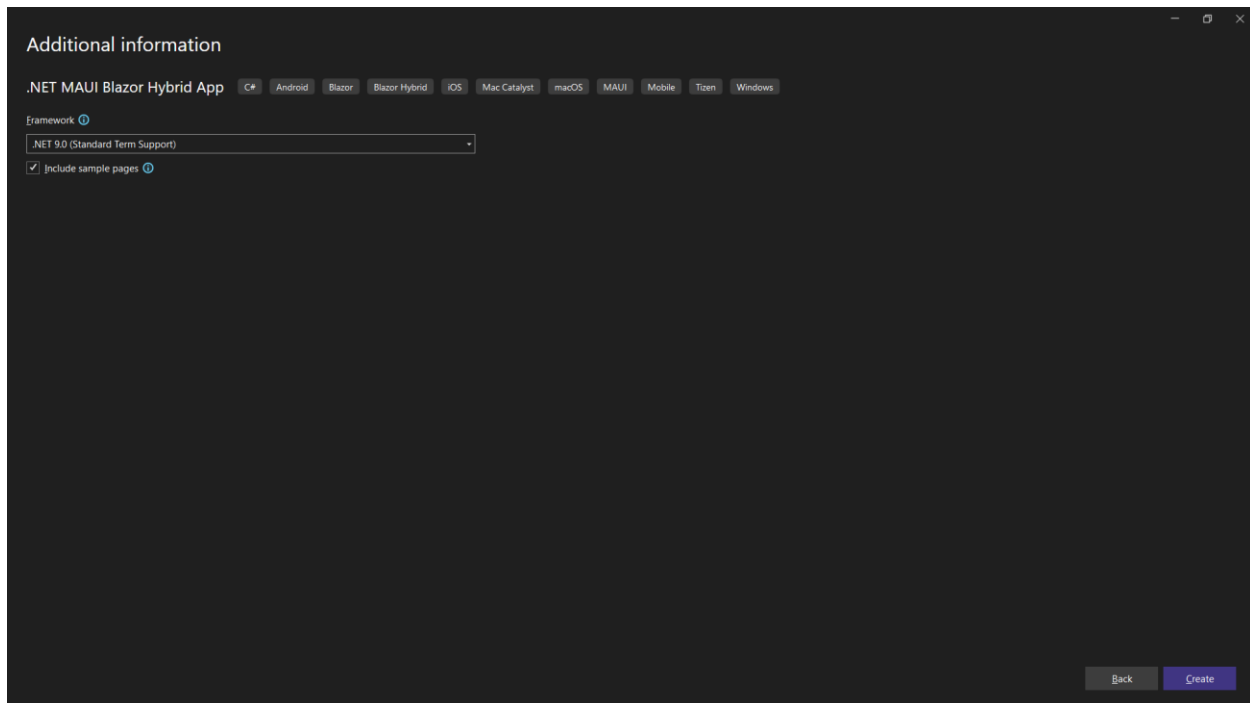


Figure 4 Selecting framework

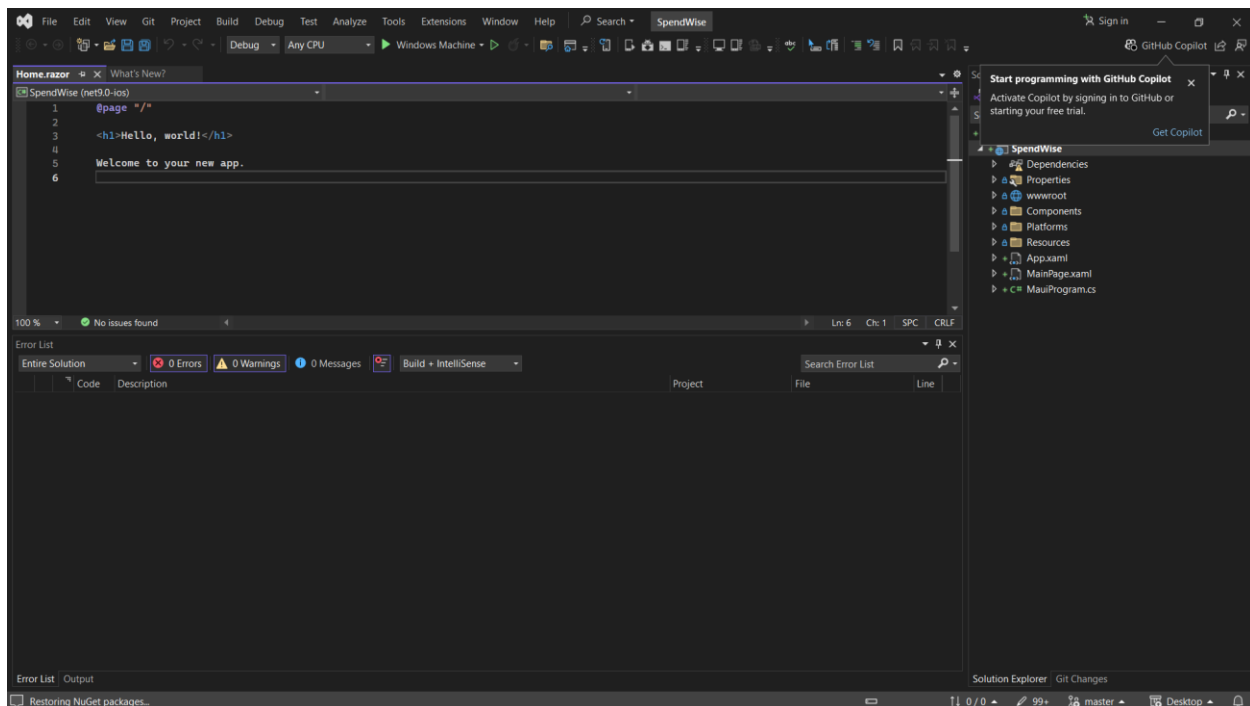


Figure 5 Project successfully created

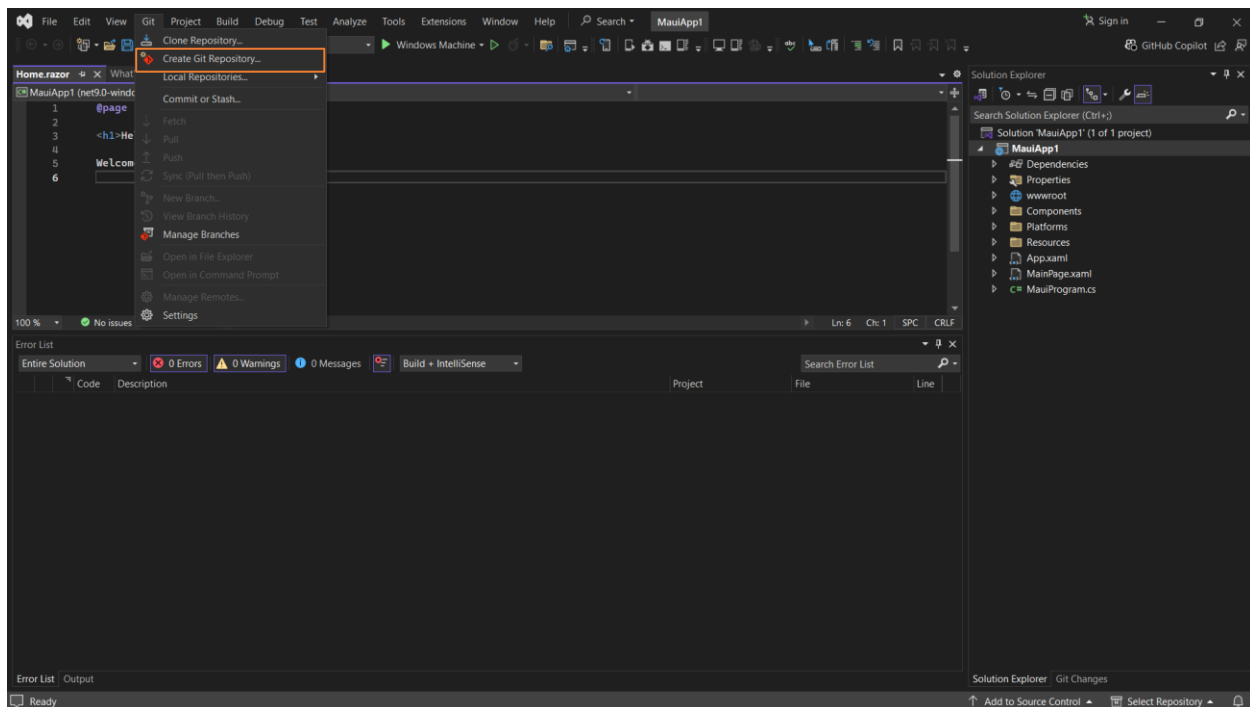


Figure 6 Creating Repository

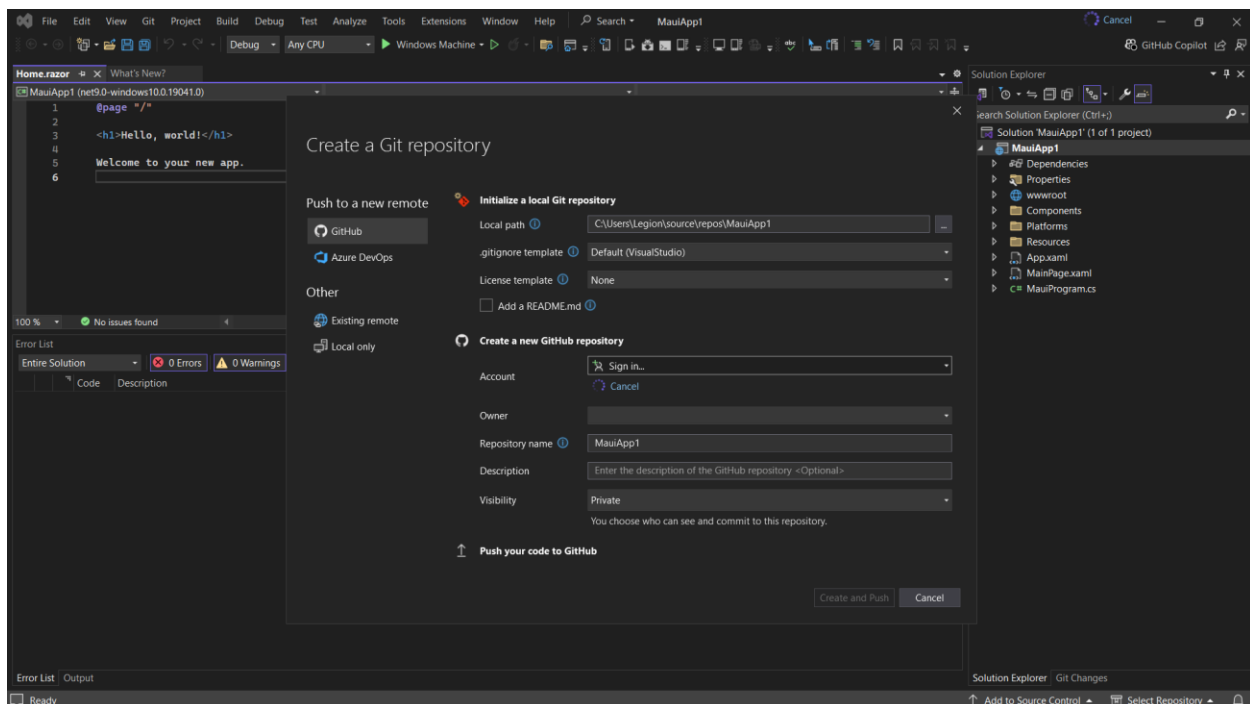


Figure 7 Signing in

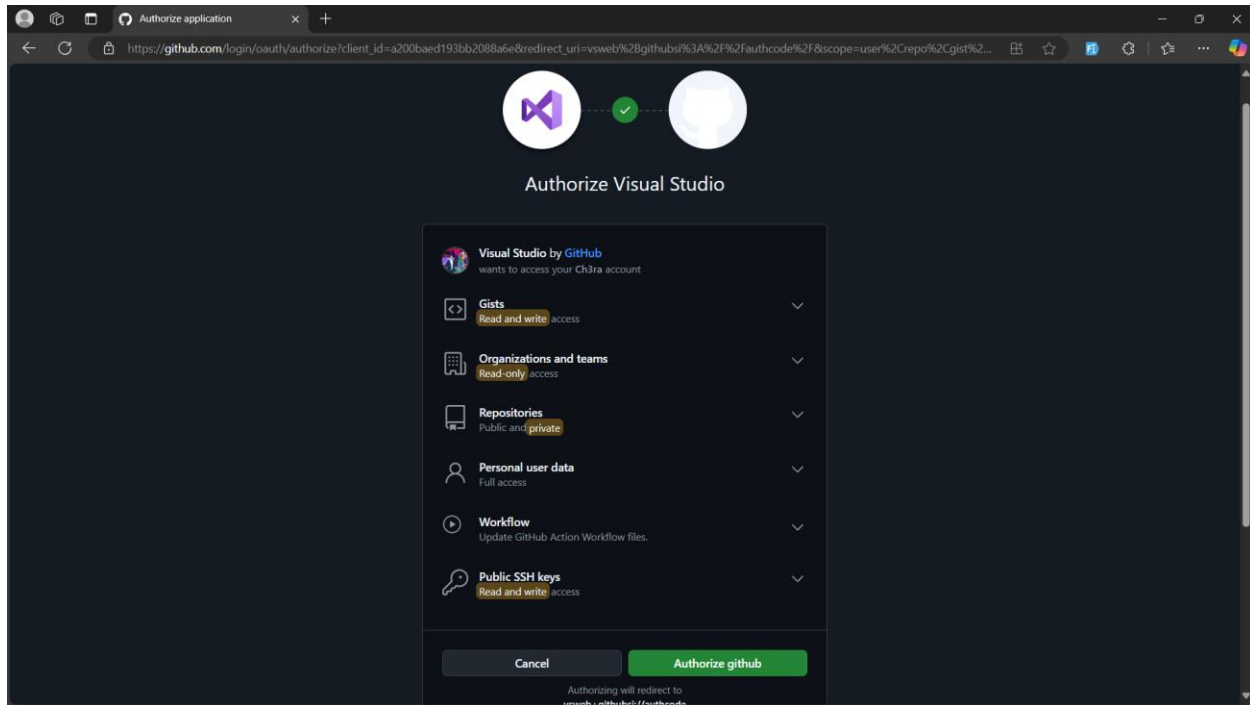


Figure 8 Connecting visual studio folder with github

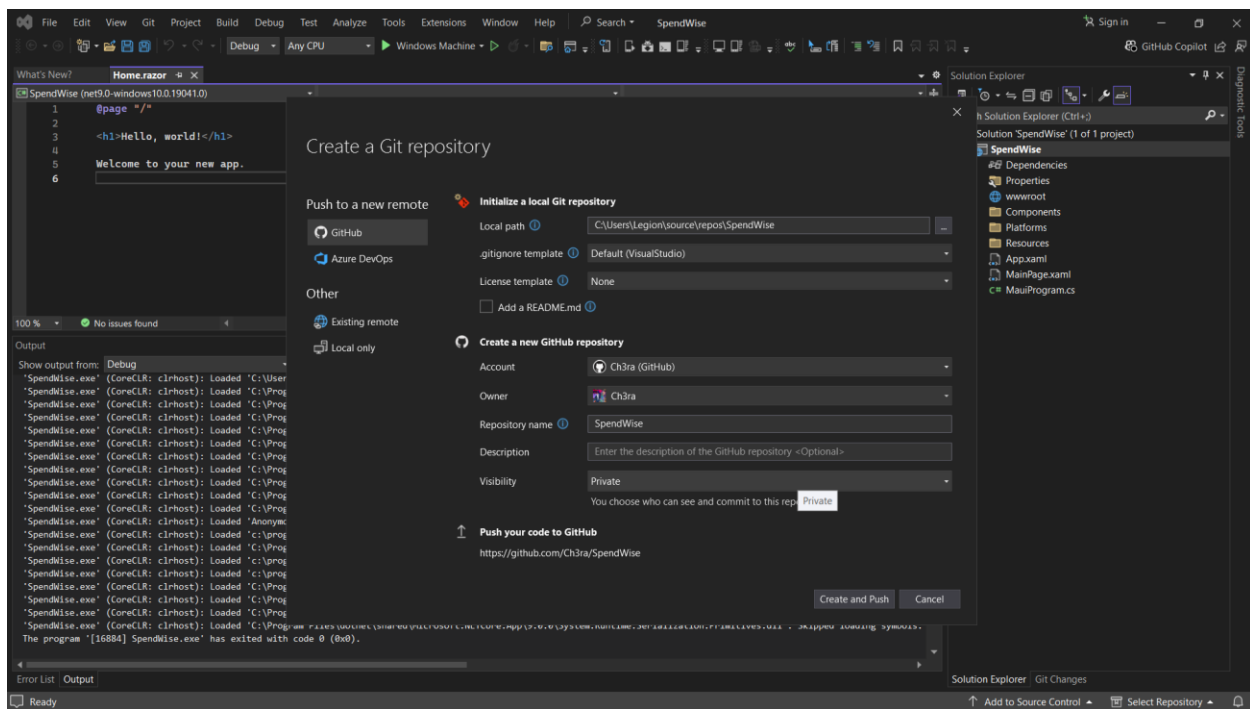


Figure 9 Connected successfully

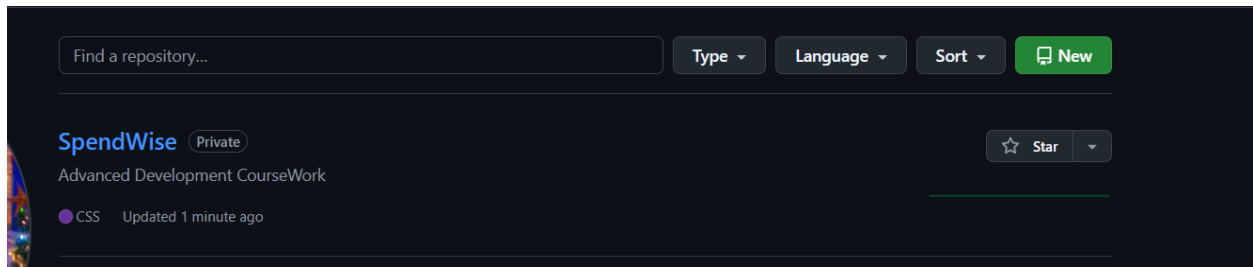


Figure 10 Github profile after creation

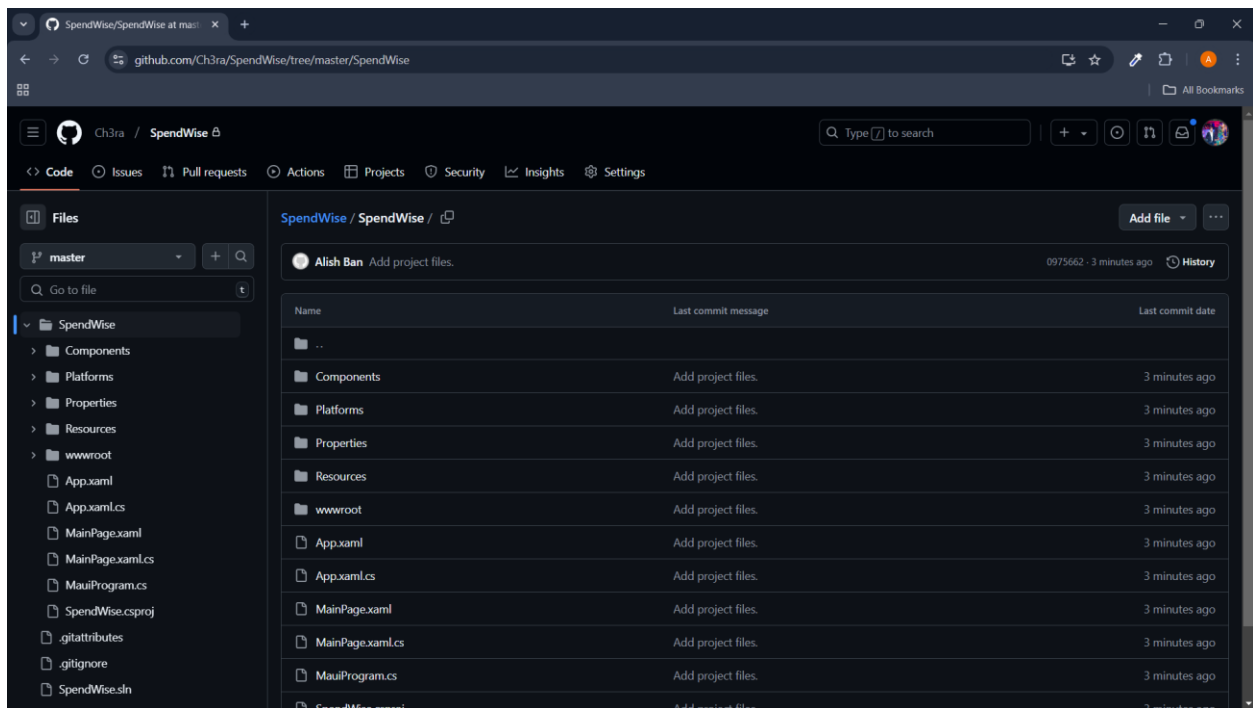


Figure 11 Github repository

Link to my GitHub: <https://github.com/Ch3ra/SpendWise>

3. UI Design (Wireframe)

A wireframe represents the fundamental blueprint of a user interface, illustrating the structural framework, content arrangement, and functional elements devoid of aesthetic embellishments such as color schemes or imagery. This tool facilitates the conceptualization of ideas, streamlining user flow planning, and forms the cornerstone of the final design. As noted by LeadOrigin (2023), I will employ Figma for this process—a cloud-based design platform distinguished by its robust real-time collaboration, intuitive user experience, and advanced features, rendering it an optimal choice for crafting and iterating wireframes with precision and efficiency.




Figure 12 Figma (*miro, 2024*)

3.1 Register

The Registration Page of the SpendWise project enables users to create an account by entering key details such as a username, account number, preferred currency, and password. Additionally, users are required to specify their chosen currency for budgeting purposes. This page ensures secure registration by implementing robust validation checks and establishes a solid foundation for tailored expense tracking.

SpendWise



Register your Account

Username

Email

Preferred currency

Account Number

Password

Have an account? [Login](#)

Figure 13 Register page of my project SpendWise

3.2 Login

The Login Page provides a secure authentication process to safeguard users' sensitive financial information from unauthorized access. Upon successful authentication, users gain access to features such as expense tracking, budgeting tools, and debt management, ensuring a protected and efficient user experience..

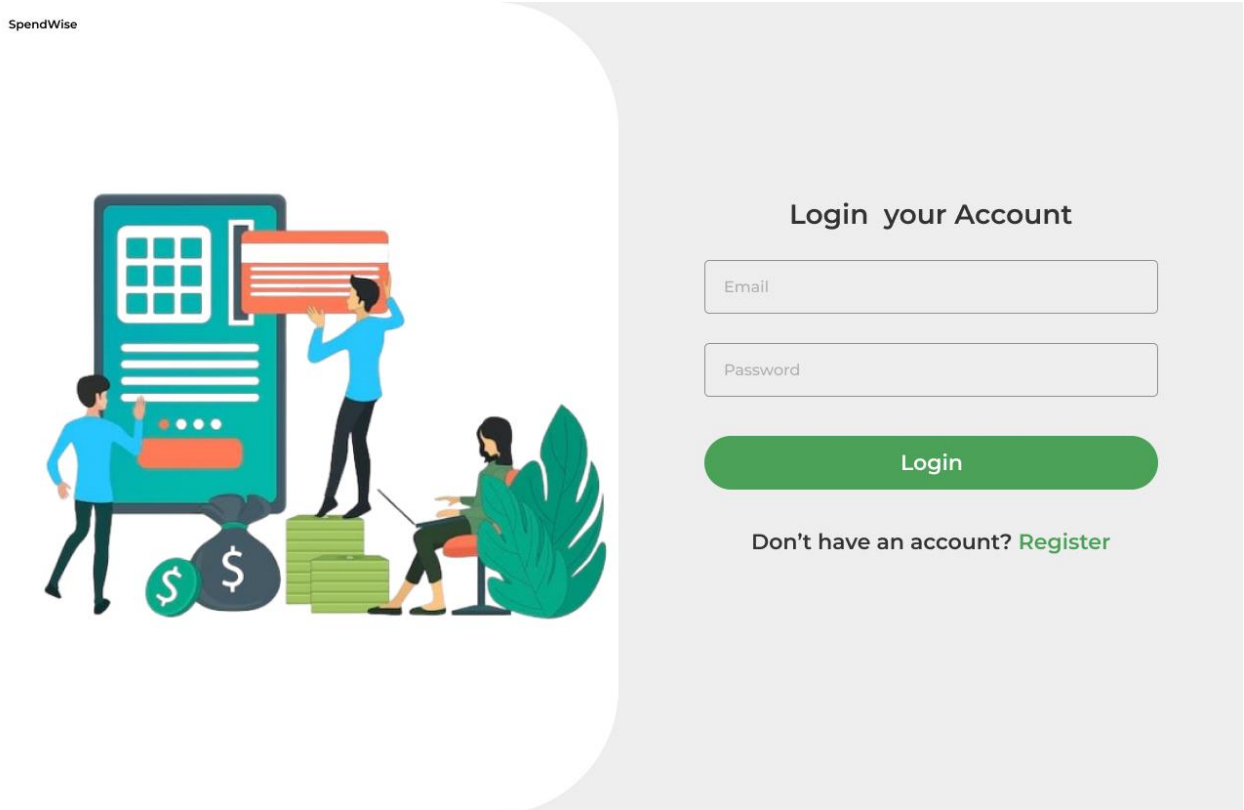


Figure 14 Login page of my project SpendWise

3.3 Dashboard

The Dashboard functions as the primary hub of the SpendWise application, offering users a concise overview of their financial activities. It highlights essential statistics, including total inflows, outflows, debts, cleared debts, and outstanding debts. Users can also review pending debts, filter data based on specific date ranges, and analyze the top five highest or lowest transactions to enhance expense management effectively.

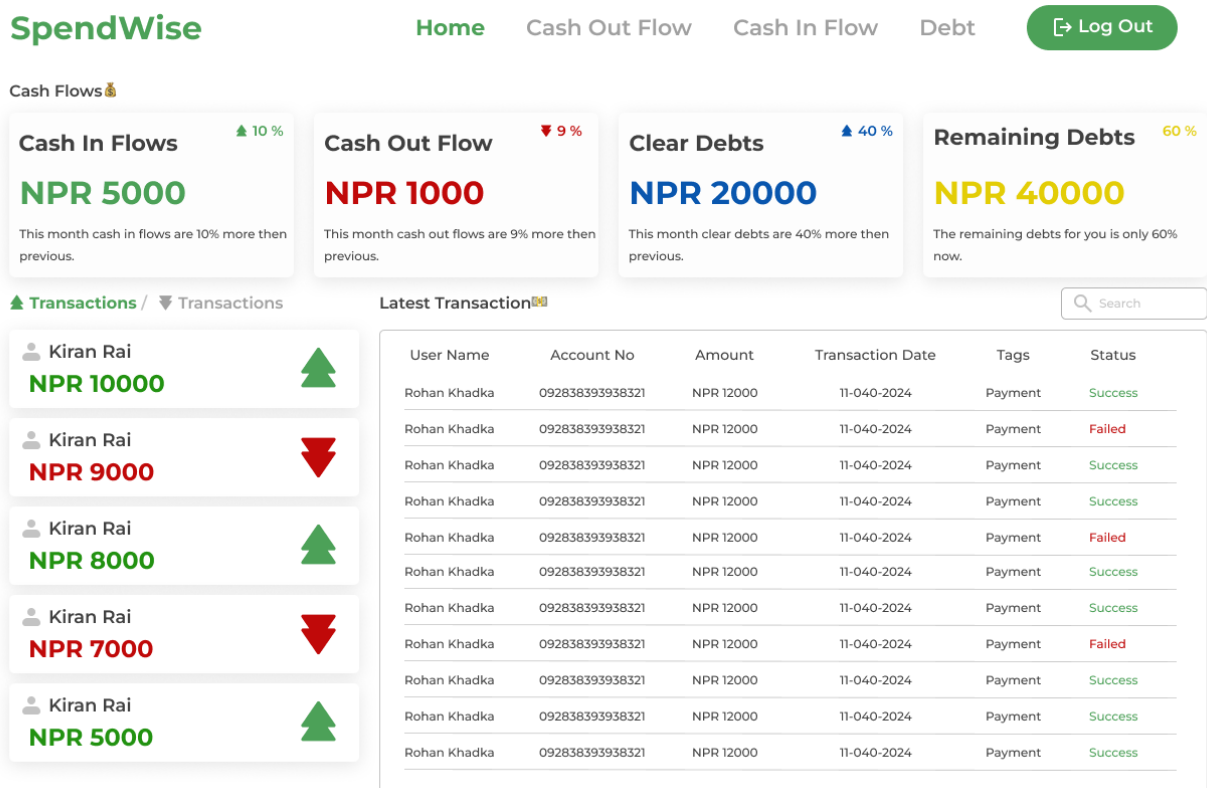


Figure 15 Dashboard of my project SpendWise

3.4 CashIn

The Cash In Page in the SpendWise project is designed to allow users to record and manage their income transactions efficiently. It focuses on capturing essential details of cash inflows to keep track of all income sources.

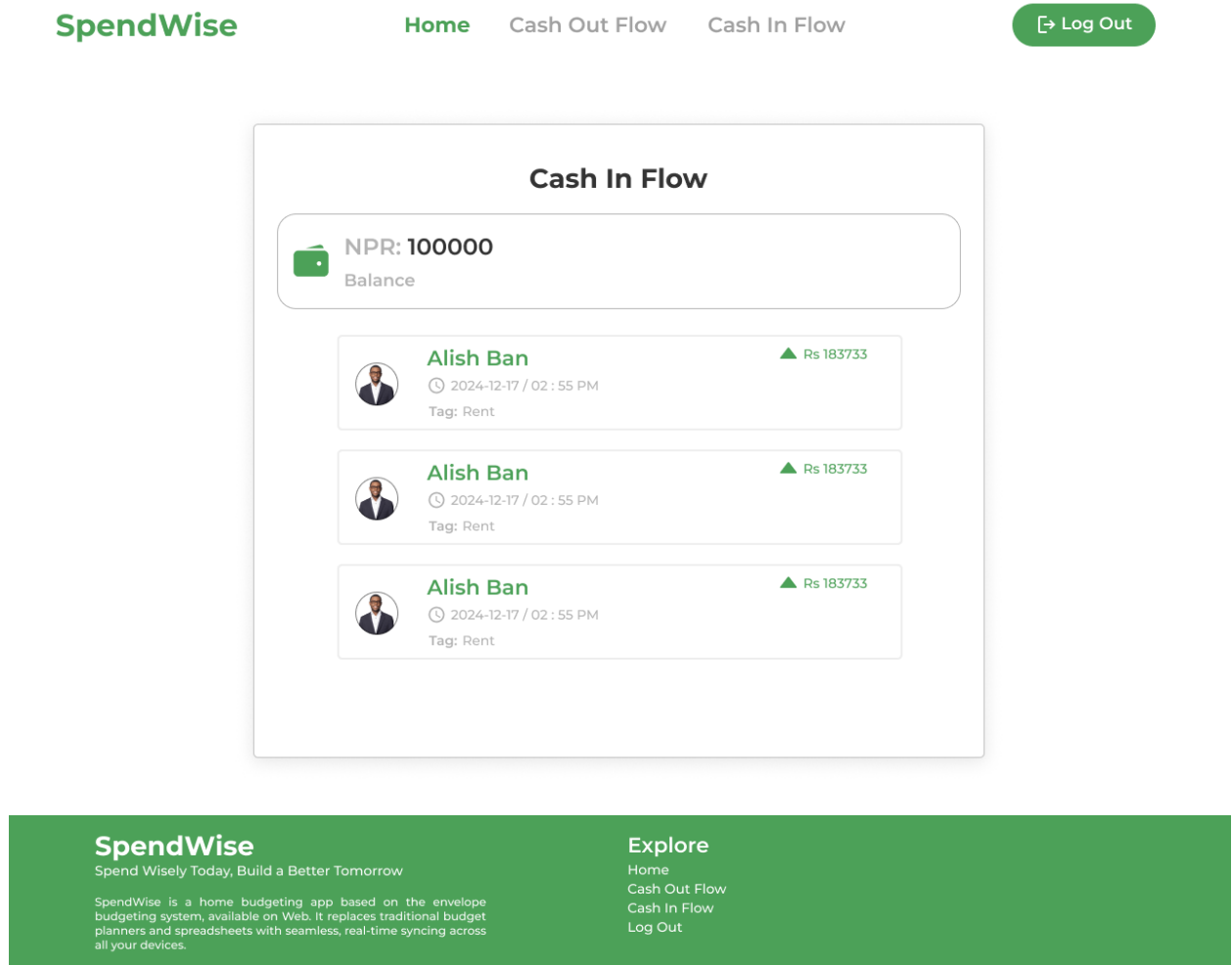



Figure 16 CashIn Flow page of my project SpenedWise

3.4 CashOut Flow

The Cash In Page within the SpendWise project is specifically designed to streamline the recording and management of income transactions. It emphasizes capturing critical details of cash inflows to effectively monitor and track all sources of income.

SpendWise[Home](#)[Cash Out Flow](#)[Cash In Flow](#)[Log Out](#)




Cash Out Flow

Username


Account No

Amount

Transaction Date

 Select Date

Tags

Select Tags 

Remarks

[Pay Now](#)

SpendWise

Spend Wisely Today, Build a Better Tomorrow

SpendWise is a home budgeting app based on the envelope budgeting system, available on Web. It replaces traditional budget planners and spreadsheets with seamless, real-time syncing across all your devices.

Explore

- Home
- Cash Out Flow
- Cash In Flow
- Log Out




Figure 17 CashhOut Flow page of my project SpendWise

4. Data/Entity Modeling

Data or Entity Modeling is a crucial aspect of designing any application, including the SpendWise project. It involves structuring and defining the data objects (entities) that the application will utilize and understanding the relationships between these entities (Visual Paradigm, 2024).

Key concepts of Data/Entity Modeling are given below:

❖ **Entities**

An entity refers to a real-world object, concept, or thing for which data is stored and systematically organized. In databases, an entity is typically represented as a table, where rows (or records) correspond to specific instances of the entity, and columns define its attributes or properties. For instance, a "User" entity might include attributes such as UserID, Name, Email, and Age, with each row in the table capturing details of an individual user.

(Ellis, 2023).

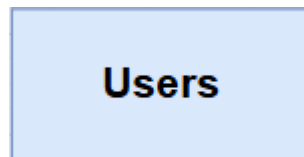


Figure 18 Entity name Users

❖ Fields/Attributes

Attributes, also known as fields, are the defining properties or characteristics of an entity that outline its structure and store its data. In databases, attributes are represented as columns in a table, where each column holds a specific type of information about the entity. For instance, a "Product" entity might include attributes like ProductID, Name, Price, and Category, which describe various aspects of a product. In object-oriented programming, attributes are represented as fields or properties within a class, encapsulating the data associated with an object. (PHOENIXNAP, 2021).

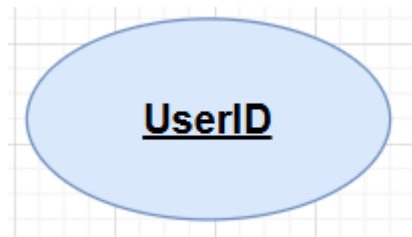


Figure 19 Attributes name UserID

❖ Relationships

Relationships describe the connections or interactions between entities. In databases, they establish logical associations between tables, such as one-to-one, one-to-many, or many-to-many relationships. For instance, in an e-commerce database, a "Customer" entity might have a one-to-many relationship with an "Order" entity, indicating that one customer can place multiple orders. In object-oriented programming, relationships are represented through references or associations between classes, enabling entities to collaborate in modeling complex systems. (Orakzai, 2023).

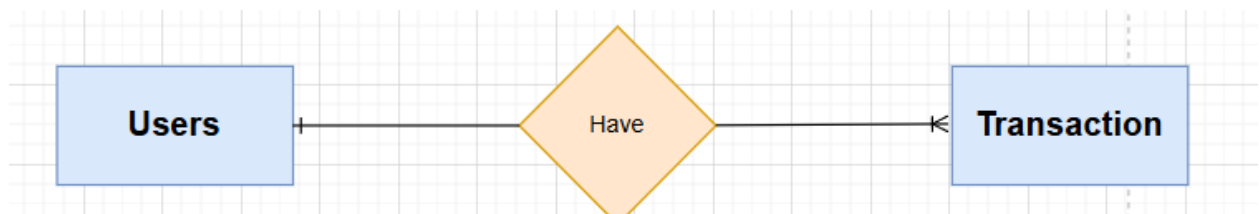


Figure 20 Relationship between the users and transaction

4.1 Entity Relation Diagram

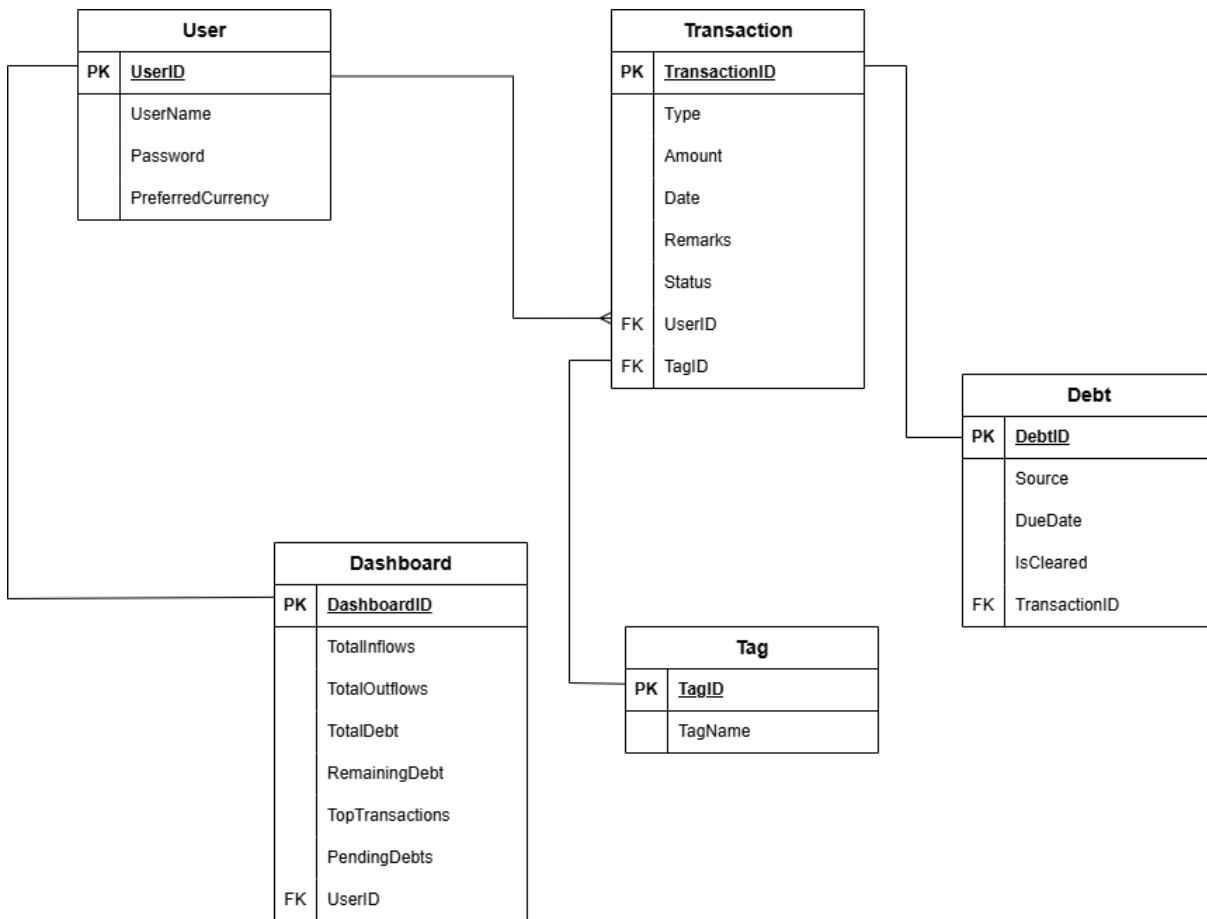


Figure 21 ERD of SpendWise

5. Technology Stack

A technology stack, or tech stack, is essentially the combination of tools, frameworks, libraries, programming languages, and software employed to develop and operate a project or application. It typically includes the frontend, responsible for managing the user interface (e.g., React, Angular); the backend, which handles server-side logic (e.g., Node.js, .NET); the database, used for data storage and management (e.g., MySQL, MongoDB); and development and deployment tools, such as Git and Docker, to streamline the workflow effectively (Scale, 2024). In essence, a tech stack serves as the foundation or "ingredients" required to create and maintain a software application (Nicole Abramowski, 2023).

Technology stack are Given below:

❖ **Framework:** MAUI Blazor Hybrid.

MAUI Blazor Hybrid is a versatile framework that integrates .NET MAUI (Multi-platform App UI) with Blazor to create cross-platform applications. It enables developers to utilize Blazor components and Razor syntax for designing user interfaces while harnessing .NET MAUI's access to native device functionalities. This hybrid approach facilitates the development of web-based user interfaces with native application capabilities for platforms like Android, iOS, Windows, and macOS, all from a single codebase. (Rastegarinia, 2023).



Figure 22 MAUI Blazor Hybrid Logo (Rastegarinia, 2023)

❖ **External Libraries:** Newtonsoft.Json.

Newtonsoft.Json, also known as Json.NET, is a widely used open-source .NET library for parsing, serializing, and deserializing JSON data. It simplifies the conversion of .NET objects to JSON format and vice versa, offering fast and flexible JSON handling. Commonly utilized in web APIs, it provides features like LINQ to JSON, custom converters, and high performance, making it a staple in .NET applications for managing JSON data effectively.

(Json.NET Documentation, 2024).

❖ **Persistence Mechanism:** File handling using JSON.

File handling with JSON entails reading and writing JSON files to store or retrieve application data. In .NET, libraries such as Newtonsoft.Json or System.Text.Json are commonly employed for serializing objects (converting them into JSON format) and deserializing JSON data (converting it back into objects). JSON files are lightweight, human-readable, and extensively utilized in modern applications for configurations, data storage, and data exchange. (Panagopoulos, 2024).

6. References

- Ellis, S., 2023. *What is an Entity in DBMS?*. [Online]
Available at: <https://www.theknowledgeacademy.com/blog/entity-in-dbms/>
[Accessed 15 12 2024].
- Json.NET Documentation, 2024. *Preserving Object REference*. [Online]
Available at: <https://www.newtonsoft.com/json/help/html/PreserveObjectReferences.htm>
[Accessed 17 11 2024].
- LeadOrigin, 2023. *What Is a Wireframe: A Foundation for Website Design*. [Online]
Available at: <https://leadorigin.com/what-is-a-wire-frame/>
[Accessed 15 11 2024].
- miro, 2024. *What is a wireframe?*. [Online]
Available at: <https://miro.com/wireframe/what-is-a-wireframe/>
[Accessed 15 11 2024].
- Nicole Abramowski, 2023. *What is a Tech Stack? A Complete Beginner's Guide*. [Online]
Available at: <https://careerfoundry.com/en/blog/web-development/what-is-a-tech-stack/>
[Accessed 16 11 2024].
- Orakzai, R. U., 2023. *What Is an Entity Relationship Diagram?*. [Online]
Available at: <https://www.baeldung.com/cs/erd>
[Accessed 15 11 2024].
- Panagopoulos, G., 2024. *Handling Circular References When Working With JSON in .NET*. [Online]
Available at: <https://code-maze.com/aspnetcore-handling-circular-references-when-working-with-json/>
[Accessed 17 11 2024].
- PHOENIXNAP, 2021. *What Is an Object-Oriented Database*. [Online]
Available at: <https://phoenixnap.com/kb/object-oriented-database>
[Accessed 15 04 2024].
- Rastegarinia, M. H., 2023. *.NET MAUI Blazor - Best practices for Desktop UI*. [Online]
Available at: <https://dev.to/mhrastegari/net-maui-blazor-best-practices-for-desktop-ui-4peo>
[Accessed 17 11 2024].
- Scale, F., 2024. *Top 8 Tech Stacks: Choosing the Right Tech Stack*. [Online]
Available at: <https://fullscale.io/blog/top-5-tech-stacks/>
[Accessed 26 11 2024].
- Visual Paradigm, 2024. *What is Entity Relationship Diagram (ERD)?*. [Online]
Available at: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/;WWWSESSIONID=04D06089560327C98CC704FDA0FA9D07.www1>
[Accessed 15 11 2024].

7. Appendix

12/19/24, 11:10 AM

22072045 AlishBan.docx

Originality report

STUDENT NAME

ALISH BAN

FILE NAME

22072045 AlishBan.docx

REPORT CREATED

Dec 19, 2024

Summary

| | | |
|-----------------------|---|------|
| Flagged passages | 3 | 4% |
| Cited/quoted passages | 1 | 0.7% |

Web matches

| | | |
|-------------------|---|------|
| cliffsnotes.com | 1 | 3% |
| coursera.org | 1 | 0.8% |
| revelo.com | 1 | 0.7% |
| sciencedirect.com | 1 | 0.7% |

in an e-commerce database, a "Customer" entity might have a one-to-many **relationship** with an "Order" entity, indicating that **one customer can place multiple orders**

[Top web match](#)

E-commerce platforms: **In an e-commerce database**, the **relationship** between customers and orders is typically **one-to-many**, as a single **customer can place multiple orders** over time.

Database Cardinality: A Brief Overview - Coursera <https://www.coursera.org/articles/cardinality>

3 of 4 passages

Student passage FLAGGED

A technology stack, or tech stack, is essentially the combination of tools, **frameworks**, **libraries**, programming languages, **and** software employed **to develop** and operate a project or **application**

[Top web match](#)

A technology stack, or tech stack, combines programming languages, **frameworks**, **libraries**, **and** other tools developers use **to develop** an **application** or website.

What is a Tech Stack? Meaning, Benefits, and Best Use Cases <https://www.revelo.com/blog/what-is-a-tech-stack>

4 of 4 passages

Student passage CITED

This hybrid approach facilitates **the development of** web-based user interfaces with native application capabilities **for platforms like Android, iOS, Windows, and macOS**, all **from a single**

[Top web match](#)

This framework allows **the development of** mobile applications **for** different **platforms like Android, iOS, Windows, or MacOS from a single** code base. In Flutter, all applications are.

Comparative analysis of mobile application

Frameworks <https://www.sciencedirect.com/science/article/pii/S1877050924010871/pdf?md5=26099e453f2c945985b96c6ac8dd1baf&pid=1-s2.0-S1877050924010871-main.pdf>

Figure 23 Organility Report