WealthE

Finance Management and Tax Filing System

CSE408 Project

A2 Group-7

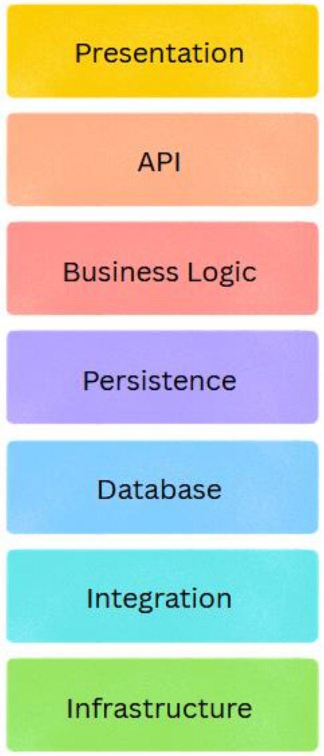
- 2005050 - Nabila Tabassum
- 2005052 - Tausif Rashid
- 2005056 - Azmal Karim

Layered Architecture

- Separation of concerns among each layer
- Scalability and reusability for individual layers
- Suitable implementation for a monolithic service

Layer Overview

1. Presentation Layer
2. API Layer
3. Business Logic Layer
4. Persistence Layer
5. Database Layer
6. Integration Layer
7. Infrastructure Layer



1. Presentation Layer

Purpose:

- Provides user-friendly interface for taxpayers, and administrators to interact with the system.
- It ensures accessibility and securely routes user inputs to backend services for accurate processing.

1. Presentation Layer

Components:

Taxpayer Interface

- Personal Information
- Income UI
- Expense UI
- Asset-Liability UI
- Investments UI
- Tax estimation and Rebate Dashboard
- Tax Return UI
- Tax History dashboard

1. Presentation Layer

Components:

Admin Interface

- Admin Information UI
- Rules Portal
- Files and Feedback UI

1. Presentation Layer

Tech Stack:

- Reactjs
- Tailwind CSS
- Shadcn UI

Tailwind CSS

React JS

shadcn/ui

2. API Layer

Purpose:

Provide connection to Presentation Layer with Business Logic Layer

Enforce authentication

2. API Layer

Components:

- CRUD Operations Service
- Input Validation
- Checking Authentication

Tech :

REST API, JWT auth

3. Business Logic Layer

Purpose:

- Implements calculations, and workflows, ensuring accurate processing of data and returns.
- Manages tax rules and logic.

3. Business Logic Layer

Components/Services:

User

- UserService : User Registration, Login, Handles Personal Information
- Financial Service : Manage financial items ie. Income, expenses, assets, investments etc.
- TaxService: Calculate and Estimate Tax, View Rebate Plans, Return Submission
- PaymentService : Ensure Tax payment validation and coordination with the Integration Layer.
- HistoryService : View Previous Tax and Financial History

3. Business Logic Layer

Components/Services:

Admin

- Rules Management Service : Manage and Add Different Tax Rules such as, income slabs, Rebate rules, Minimum Tax by Zone etc.
- FeedBack Service : Provides service for Viewing and giving feedback to users on Return Submissions

3. Business Logic Layer

Tech Stack:

- Spring Boot
- Java

Ensures Enterprise Standard

4. Persistence Layer

Purpose:

- Abstracts database interactions from the business layer
- Provides a clean interface for CRUD operation on database

4. Persistence Layer

Components:

- Data Access Objects (DAO) : handle CRUD operations and complex queries

  User : PersonalInfo, TaxInfo

  Financial Item: Income, Expense, Asset, Liability, Investment

  Tax: Tax Return, Payment

  Admin : Rules

Tech Stack:

Spring Data JPA (Java Persistence API)

5. Database Layer

Purpose:

- Data Storage : Store rules, User information, income/expense records
- Data Retrieval : Efficient query of data
- Integrity : Ensure data accuracy, consistency
- Relations : Support complex relation

5. Database Layer

Components:

- Relational Database : Store UserInfo, TaxInfo, Rules etc data in a structured manner.

Tech Stack :

PostgreSQL

6. Integration Layer

Purpose:

- Facilitates communication with external services and APIs, such as payment gateways, sms verification, and email notification.

6. Integration Layer

Components:

- Payment Gateway Integration: Manages interaction with third-party services eg. bKash .
  Tech Stack : bkash API(mock)
- Email Notification: Send Submission Confirmation, Login Alert.
  Tech Stack : SendGrid
- External API Integration : NBR api (mock) for TIN, NID Number verification
- AI features: Google AI studio

SendGrid

Google AI Studio

7. Infrastructure Layer

Purpose:

- Provides foundational support for hosting, scalability and deployment.
- Ensure smooth performance in various environments

7. Infrastructure Layer

Component:

- Hosting : Runs the application on a scalable cloud service.
  Tech Stack : Azure VM
- Containerization : Ensures consistent deployment across environments.
  Tech Stack : Docker
- Storage and Data Backup : Storage for application and regular backup for Disaster Recovery.
  Tech Stack : Azure PostgreSQL
- Monitoring and Logging : Tracks System performance, identifies bottlenecks, logs critical events.
  Tech Stack : Azure Monitor + Application Insights

7. Infrastructure Layer

Component:

- CI/CD : Automates the build, test, and deployment processes.
  Tech Stack : Github Actions
- Load Balancing : Scale servers dynamically based on traffic .
  Tech Stack : NGINX, Kubernetes
- Security : Ensure https, protection from DDOS attacks

  Tech Stack : Cloudflare + Spring config, azure for TLS certificate

Thank You!