**Project Report**

on

**Habit Tracker**

By

**Jawade Ajay Latari**

Under the Guidance of

**Prof. Bharati Ramageri**

**Master of Computer Application**

**P.E. Society’s**

**Modern Institute of Business Studies (Autonomous), Nigdi, Pune-44**

**Savitribai Phule Pune University**

**2024 - 2025**

**Progressive Education Society’s**

**Modern Institute of Business Studies (Autonomous)**

**Nigdi ,Pune-44**

**Certificate**

This is to certify that the Mini Project entitled **Daily Expense Manager** is submitted by **Jawade Ajay Latari** a bonafide student of this institute, studying for a Master of Computer Application, has successfully carried out MCA semester-II Mini Project documentation. This project report is submitted in partial fulfilment of **M.C.A. Semester I** curriculum as per the **Savitribai Phule Pune University** norms**.**

**Dr. Maithili Arjunwadkar Internal Guide External Examiner**

**Director**

**PES MIBS**

**ACKNOWLEDGEMENT**

It is my proud privilege to express my profound gratitude to the entire management of Progressive Education Society’s Modern Institute of Business Studies and teachers of the institute for providing me with the opportunity to avail the excellent facilities and infrastructure of the institute. The knowledge and values inculcated have proved to be of immense help to my career.

I am grateful to **Prof.** **Dr. Maithili Arjunwadkar (Director, MIBS)** for their astute guidance, constant encouragement and sincere support for this project work.

I express my sincere gratitude to Prof. Bharati Ramageri for their inspiration, constructive suggestion, mastermind analysis and affectionate guidance in my work, without which this project work completion would have been impossible for me. Sincere thanks to all my seniors and colleagues at the company for their support and assistance throughout the project.

I would like to thank all the faculties for providing me with an opportunity to pursue my industrial training, as it is an important part of the MCA course.

I feel proud and privileged in expressing my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavour if I do not place my acknowledgement.

**Name of Student**

**Ajay Latari Jawade**

**Index**

|  |  |  |
| --- | --- | --- |
| Chapter | Description | Page Number |
| 1 | INTRODUCTION |  |
| 1.1 | Introduction of project | 6 |
| 1.2 | Existing System and Need for System | 7 |
| 1.3 | Scope of Work | 8 |
| 1.4 | Operating Environment - Hardware and Software | 9 |
| 1.5 | Detail Description of Technology Used | 10 |
| 2. | PROPOSED SYSTEMS |  |
| 2.1 | Objectives of System | 11 |
| 2.2 | Proposed System | 12 |
| 2.3 | User Requirements | 13 |
| 3 | ANALYSIS AND DESIGN |  |
| 3.1 | Entity Relationship Diagram ( ERD ) | 14 |
| 3.2 | Data Flow Diagram (DFD) | 15 |
| 3.3 | Object Diagram | 16 |
| 3.4 | Class Diagram | 17 |
| 3.5 | Use Case Diagrams | 18 |
| 3.6 | Activity Diagram | 19 |
| 3.7 | Collaboration Diagram | 20 |
| 3.8 | Deployment Diagram | 21 |
| 3.9 | Component Diagram | 22 |
| 3.10 | Table Design | 23 |
| 3.11 | Data Dictionary | 24 |
| 4 | Drawbacks and Limitations | 25 |
| 5 | Proposed Enhancements | 26-27 |
| 6 | Conclusion | 28 |
| 7 | Bibliography | 29 |
| 8 | ANNEXURES: |  |
| 8.1 | ANNEXURE 1 : USER INTERFACE SCREENS | 30-33 |
| 8.2 | ANNEXURE 2 : OUTPUT  REPORTS WITH DATA ( if any ) | 34 |
| 8.3 | ANNEXURE 3 :  Sample Code | 35-43 |

**Chapter 1: INTRODUCTION**

* 1. **Introduction of Project**  
     The Expense Manager is a web-based application designed to help students manage their daily financial activities effectively. By automating the recording of income and expenses, the system reduces the manual effort required for financial tracking. It enables users to categorize expenses, analyze spending patterns, and generate detailed reports, promoting better decision-making. The application is built with an intuitive interface, ensuring ease of use for students with minimal technical expertise. Designed for scalability and reliability, the system integrates secure data handling mechanisms to protect user information. Its ultimate goal is to make financial management streamlined and error-free.

**1.2 Existing System and Need for System**  
Manual systems for financial management are prone to errors, inefficiency, and data loss. Traditional methods fail to provide comprehensive insights into spending patterns or trends. Existing digital tools often lack the simplicity or customization needed for student use. The Expense Manager addresses these gaps by offering an automated, user-friendly platform. It eliminates redundancies, minimizes errors, and provides real-time access to financial data. The system is designed for accessibility, allowing users to manage their finances from anywhere with an internet connection. This need for a modern, efficient, and reliable solution led to the development of the Expense Manager.

**1.3 Scope of Work**

* **Simplified Financial Record-Keeping**: Provides tools for logging income and expenses quickly and efficiently.
* **Transaction Categorization**: Transactions are categorized into predefined segments like food, travel, and entertainment, making it easy to analyze spending patterns.
* **Financial Reports**: Students can generate comprehensive financial reports to understand their financial habits and make informed decisions.
* **Multi-User Support**: Supports both student and parent accounts, allowing parents to assign budgets and monitor expenses.
* **Web-Based Accessibility**: Accessible through a web browser, ensuring users can manage finances on any compatible device.
* **Automated Calculations**: Automatically calculates total income, expenses, and remaining balance, reducing the risk of errors.
* **User-Friendly Interface**: Features a simple and intuitive design, making it easy for students to navigate.

**1.4 Operating Environment**  
The software requirements for developing this project are as follows:

* **Front End**: JSP
* **Back End :** Servlets
* **Database**: MySQL
* **Application Server**: Apache Tomcat
* **Operating System**: Windows 7 or higher
* **Development Tool**: Eclipse IDE

**Hardware Requirements**

The hardware requirements for developing this project are as follows:

* **Processor**: Intel Core i3
* **Memory (RAM)**: 4GB
* **Storage (HDD)**: 40GB

**1.5 Detail Description of Technology Used**  
The system is built using JSP and Servlets for the front-end, offering a dynamic and interactive user experience tailored to students' financial management needs. MySQL is employed as the back-end database, providing secure, efficient, and reliable data storage, ensuring that all transactions and user information are preserved accurately. Apache Tomcat serves as the web server, seamlessly handling client-server interactions, ensuring smooth and responsive performance. Eclipse IDE is used as the development environment, enabling organized coding practices and efficient debugging.

This combination of technologies ensures the application is scalable and adaptable, making it suitable for evolving user needs. The user-friendly interface, combined with the robust backend infrastructure, provides an efficient way for students and parents to manage their financial records. The system also prioritizes performance, offering quick processing of transactions and report generation, minimizing delays.

The architecture’s modular design makes it easy to enhance or expand functionality in the future, ensuring longevity and adaptability. Together, these elements create a reliable, efficient, and accessible application for financial management.

**Chapter 2: PROPOSED SYSTEMS**

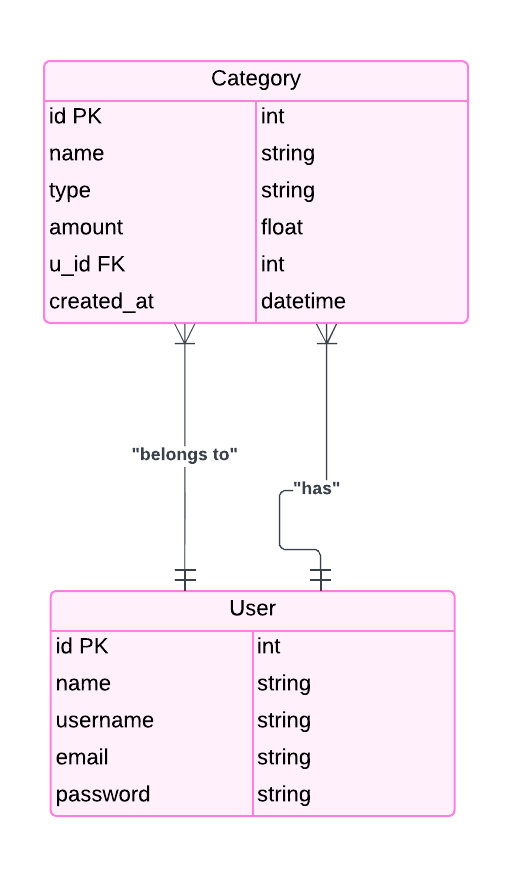
**2.1 Objectives of System**The main objective of the Expense Manager is to automate financial management processes for students. It aims to eliminate manual errors, reduce the time spent on calculations, and provide accurate financial records. The system promotes financial awareness by categorizing expenses and generating detailed reports, helping students understand their spending patterns. It ensures data security through robust back-end mechanisms and offers a simple, intuitive interface for ease of use. The overarching goal is to empower students with a reliable tool that enhances financial transparency and supports better financial decision-making.

**2.2 Proposed System**  
The proposed system is a web-based application that simplifies financial tracking for students. It allows users to log daily expenses and income without the hassle of manual record-keeping. The application automatically categorizes expenses, enabling users to analyze spending habits effortlessly. A major feature is its ability to generate customizable financial reports, which provide actionable insights into financial trends. The system is accessible via any device with an internet connection, offering convenience and flexibility. Its robust back-end ensures data security, while the user-friendly interface minimizes the learning curve, making financial management efficient and straightforward.

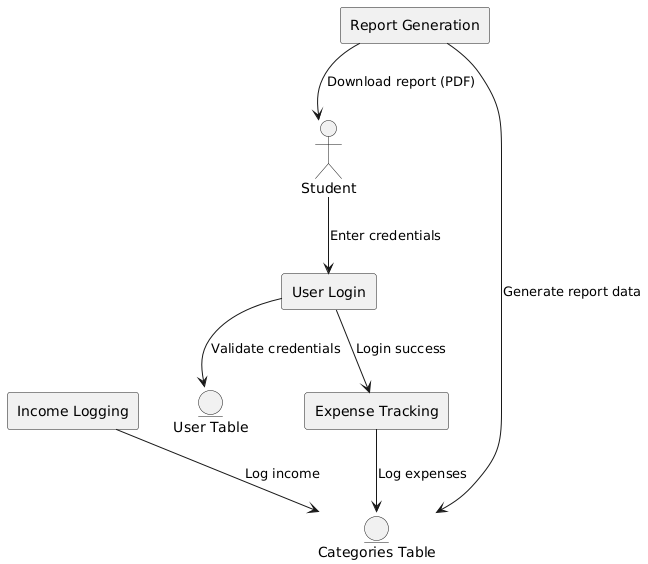
**2.3 User Requirements**Students require an easy-to-use platform for managing their finances. The Expense Manager meets this need by providing secure login access, personalized dashboards, and tools for tracking income and expenses. Users can categorize their expenses and analyze financial trends with just a few clicks. The system must also allow users to generate and download detailed financial reports, aiding in decision-making. Accessibility is key, so the platform is web-based, enabling students to use it on any internet-enabled device. Additionally, the system ensures data security and reliability, addressing the core requirements of student users.

**Chapter 3: ANALYSIS AND DESIGN**

**3.1 Entity Relationship Diagram (ERD)**

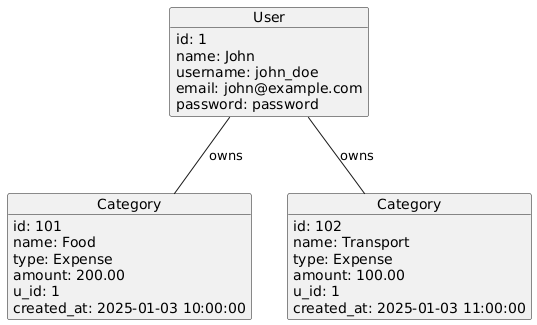


**3.2 Data Flow Diagram (DFD)**

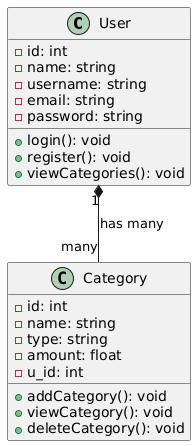


**3.2 Data Flow Diagram (DFD)**

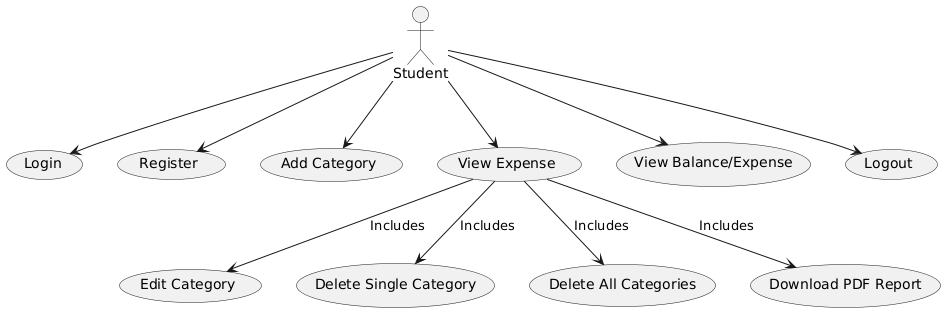
**3.3 Object Diagram**



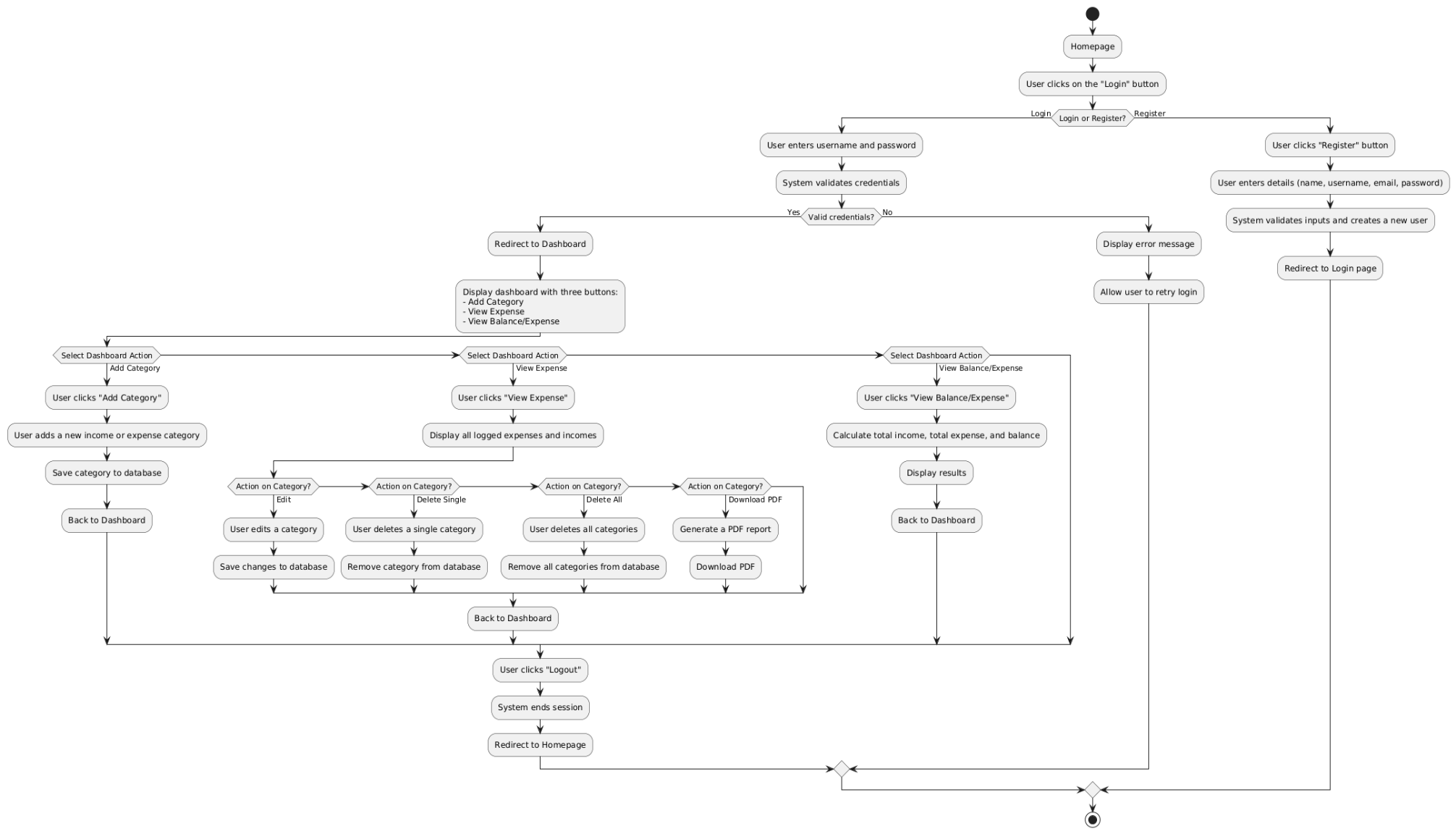
**3.4 Class Diagram**

****

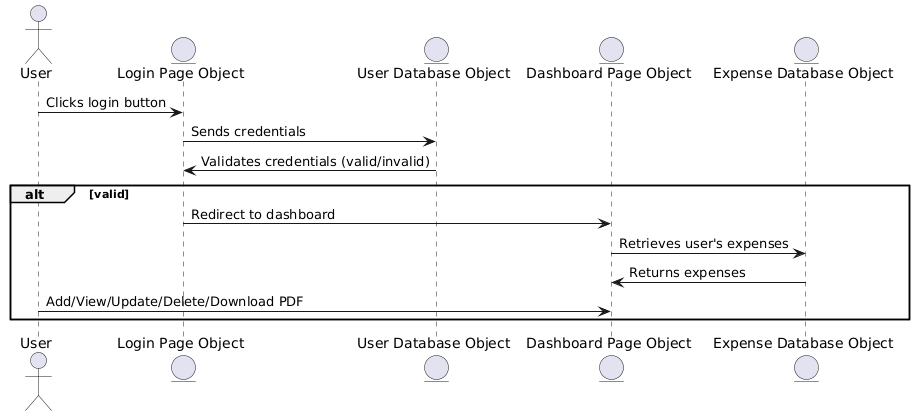
**3.5 Use Case Diagrams**



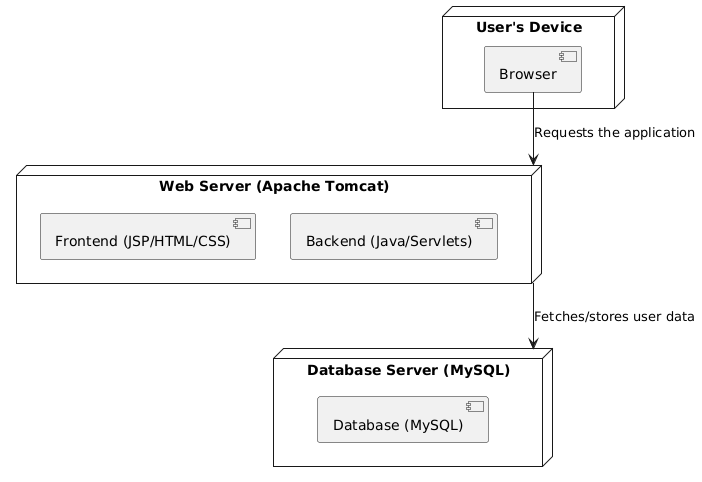
**3.6 Activity Diagram**



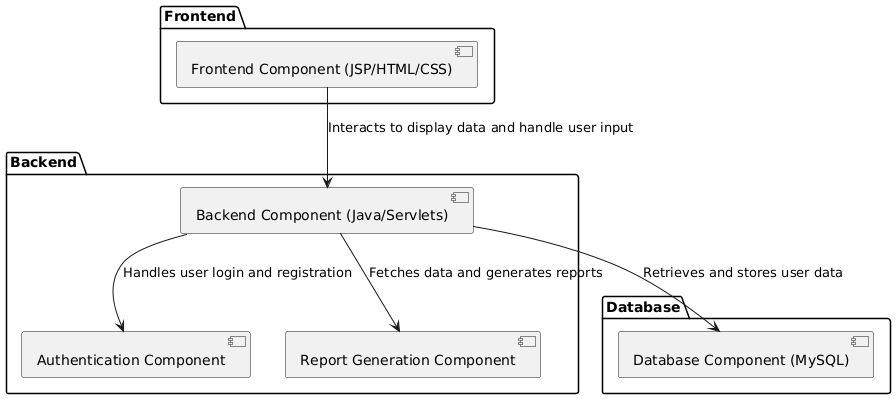
**3.7 Collaboration Diagram**



**3.8 Deployment Diagram**

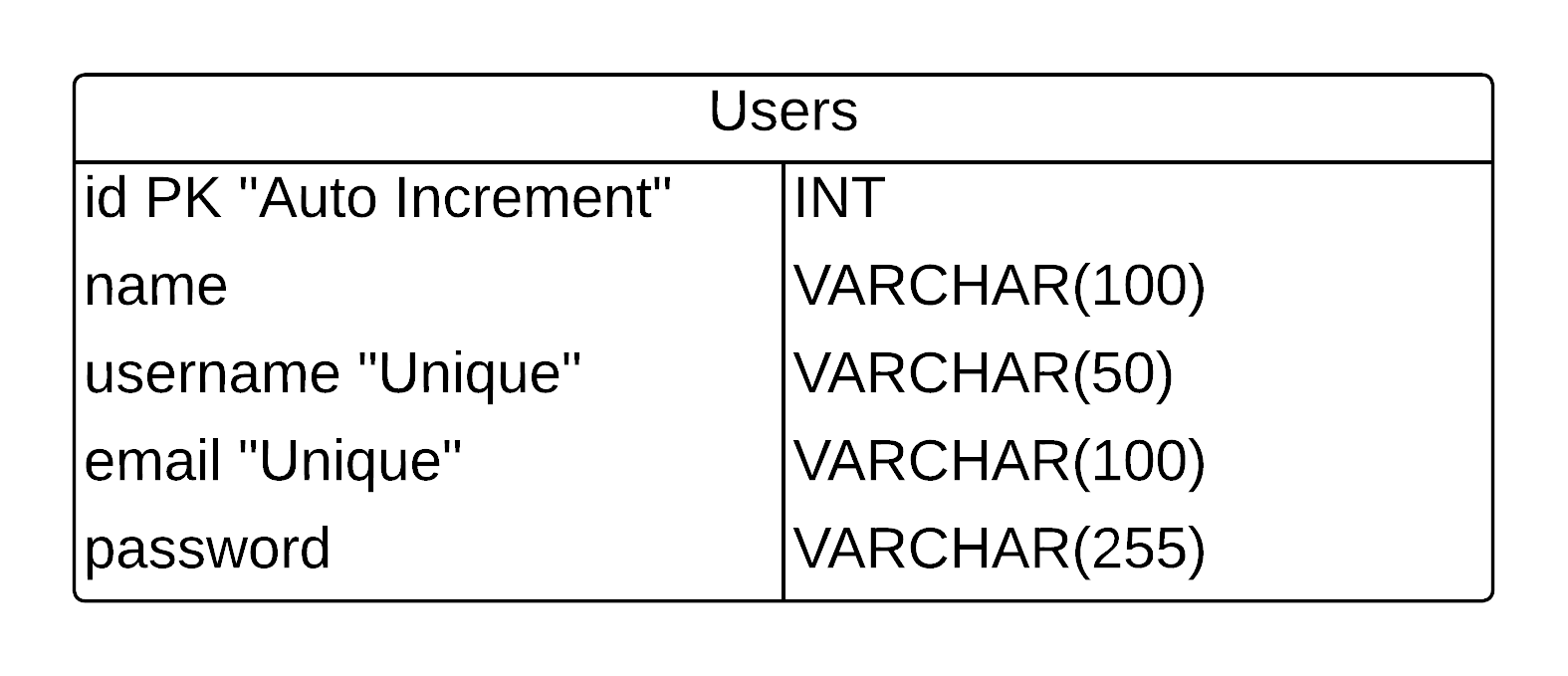
****

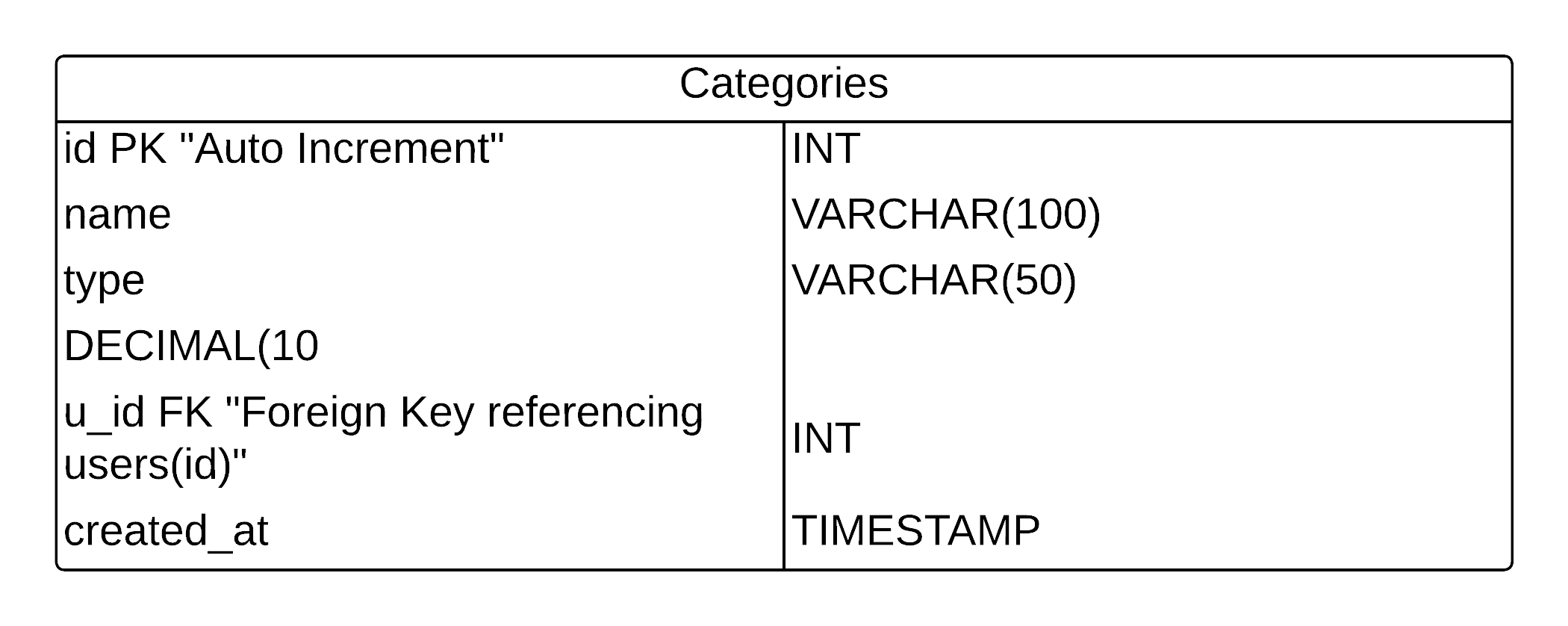
**3.9 Component Diagram**

****

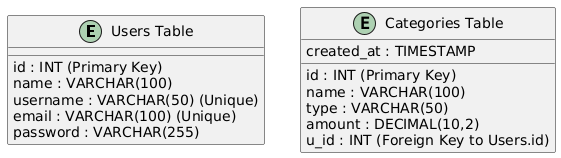
**3.10 Table Design**

**1.Users Table**



**2.Categories Table**

**3.11 Data Dictionary**



**Chapter 4: Drawbacks and Limitations**

While the Expense Tracker application offers useful features, it has several limitations that could affect its overall usability and adaptability. These drawbacks include:

1. **Limited Offline Functionality**:  
   The system requires a constant internet connection to function. This dependency on internet access may be problematic for users in areas with unstable or no connectivity, limiting its usability in such circumstances. Future versions could include offline capabilities, allowing users to track expenses without being connected to the internet.
2. **Heavily Dependent on User Input**:  
   The accuracy of the system relies heavily on users providing correct and consistent data. If users enter incorrect or incomplete information, the system’s reports and insights may become inaccurate, potentially affecting decision-making. Implementing validation checks and providing users with guidance on how to input data accurately could help mitigate this issue.
3. **Primarily Designed for Students**:  
   The current version of the application is primarily designed to cater to the needs of students, making it less adaptable for other user groups such as working professionals or families. To improve versatility, the system could be enhanced to allow customization, providing different user groups with features tailored to their specific needs.
4. **Limited Reporting and Analytics**:  
   The current reporting features are basic, and more advanced analytics and visualizations could improve the user experience. By adding graphs, trends, and forecasting features, users could gain more insightful financial summaries.
5. **Alert Messages**: The system currently doesn't have any alert or notification messages to inform users about critical actions or errors.
6. **Setting the Budget**:

The application doesn't allow parents to set a budget for their child's spending.

**Chapter 5: Proposed Enhancements**

To enhance the functionality and usability of the Expense Tracker, several improvements can be introduced in future versions. These enhancements could significantly improve the user experience, provide more value, and increase the system’s adaptability. The following proposed enhancements could be considered:

1. **Integration with Payment Gateways**:
   * **Improvement**: Integrating with popular payment gateways could automate transaction logging. This integration would allow users to track their expenses and incomes seamlessly without needing to manually enter each transaction.
   * **Benefit**: This would reduce the chances of errors, save time for users, and ensure that the expense data is updated in real-time.
2. **Mobile Application Development**:
   * **Improvement**: Developing a mobile app version of the Expense Tracker would make it more accessible. Users could track their expenses, check their budgets, and add new categories directly from their smartphones.
   * **Benefit**: A mobile app would provide greater convenience, allowing users to access the system on-the-go, enhancing the overall user experience.
3. **AI-Based Financial Insights**:
   * **Improvement**: Implementing artificial intelligence (AI) to analyze user spending habits and provide personalized financial insights could help users make more informed financial decisions. AI could detect patterns and suggest areas where users could cut costs or invest more wisely.
   * **Benefit**: Smarter, data-driven spending recommendations would empower users to manage their finances better and make informed decisions to stay within their budgets
4. **Predictive Analytics for Budgeting**:
   * **Improvement**: Introducing predictive analytics could help users forecast their future expenses based on historical data, income patterns, and spending behavior.
   * **Benefit**: Predictive budgeting would provide users with a better understanding of their future financial situation, helping them plan and
   * allocate funds more effectively.

By integrating these enhancements, the Expense Tracker could transform into a more comprehensive, user-friendly, and intelligent tool, ultimately improving financial management for its users.

**Chapter 6: Conclusion**

The Expense Manager has proven to be an effective tool for students seeking to manage their finances with ease and precision. Its user-friendly interface, coupled with essential features such as automated transaction logging, comprehensive reporting, and budget management, makes it an indispensable resource for students aiming to stay on top of their expenses. Through features like detailed income and expense tracking, students can quickly gain insights into their spending habits and identify areas for improvement.

By replacing outdated, manual financial tracking methods, the Expense Manager simplifies the process of managing finances and encourages users to become more mindful of their financial behavior. It fosters financial awareness and discipline, making it easier for students to set goals, track progress, and adjust their spending habits as necessary. These improvements lead to smarter, more informed decision-making, helping students avoid financial pitfalls and build better financial habits.

As the application evolves with future enhancements such as payment gateway integration, mobile app development, and AI-driven insights, it has the potential to become an even more powerful tool for users of all backgrounds, not just students. Overall, the Expense Manager stands as a promising solution for individuals looking to take control of their finances, offering both immediate benefits and long-term value.

**Chapter 7: Bibliography**

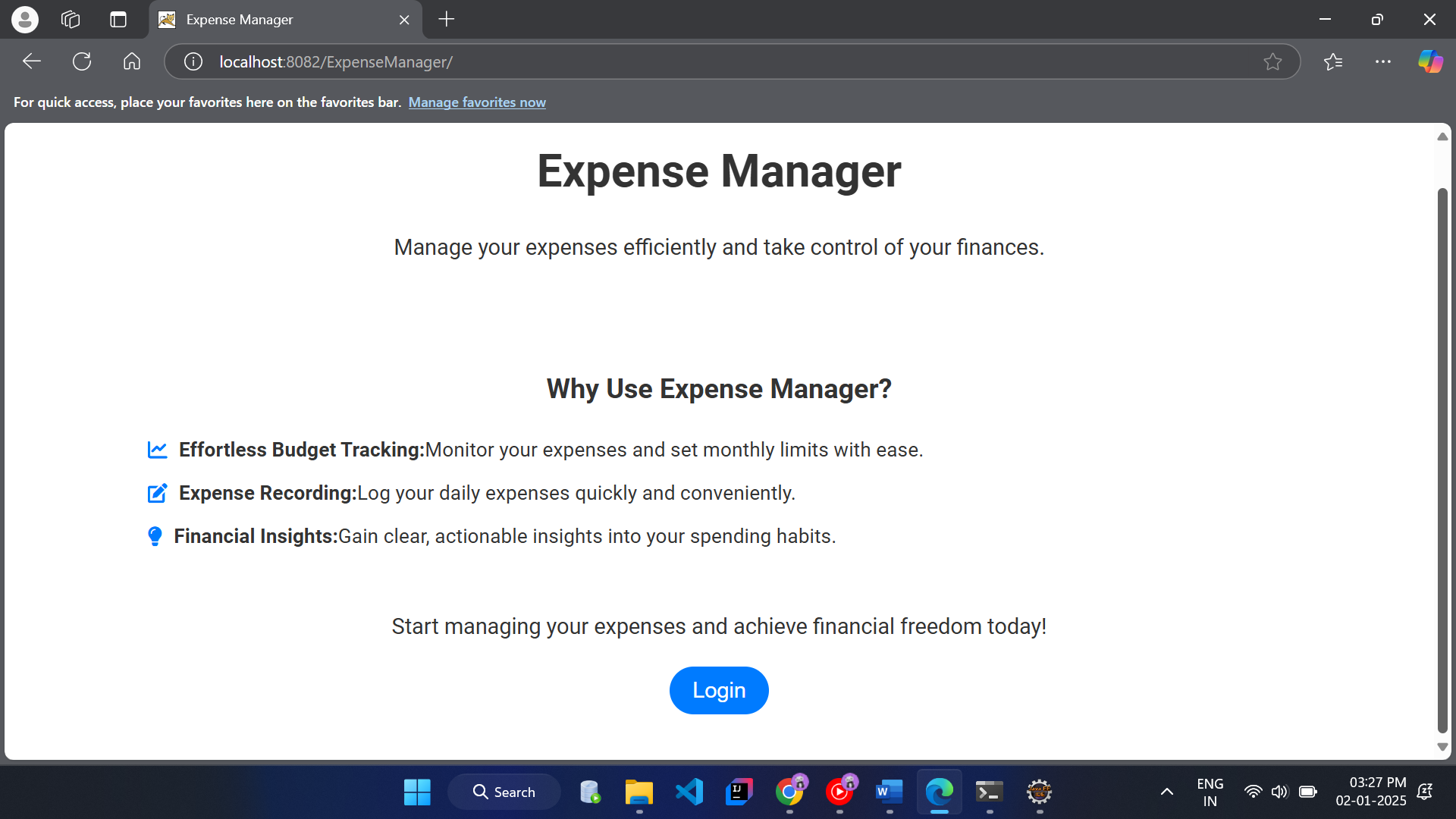
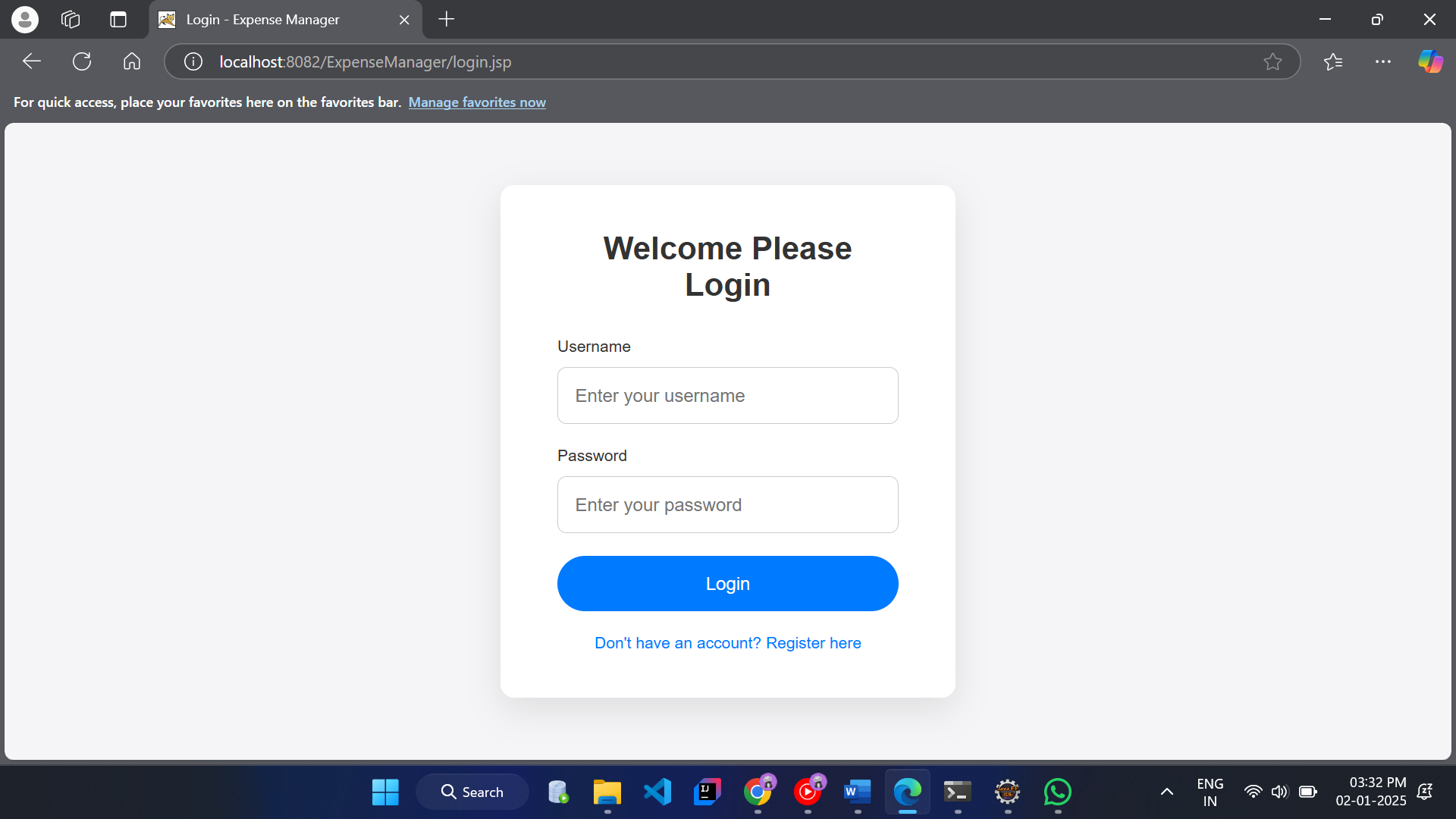
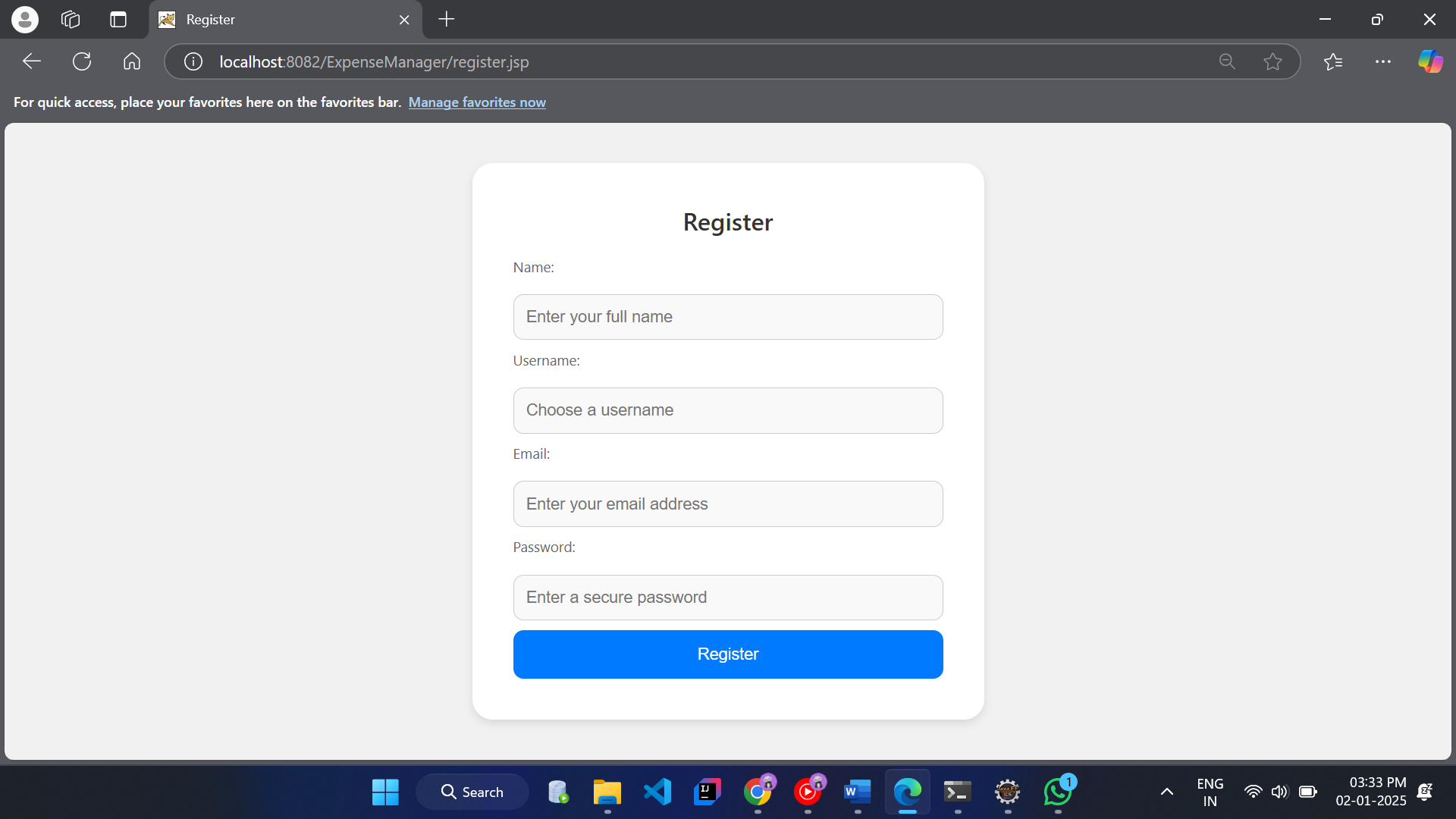
Books

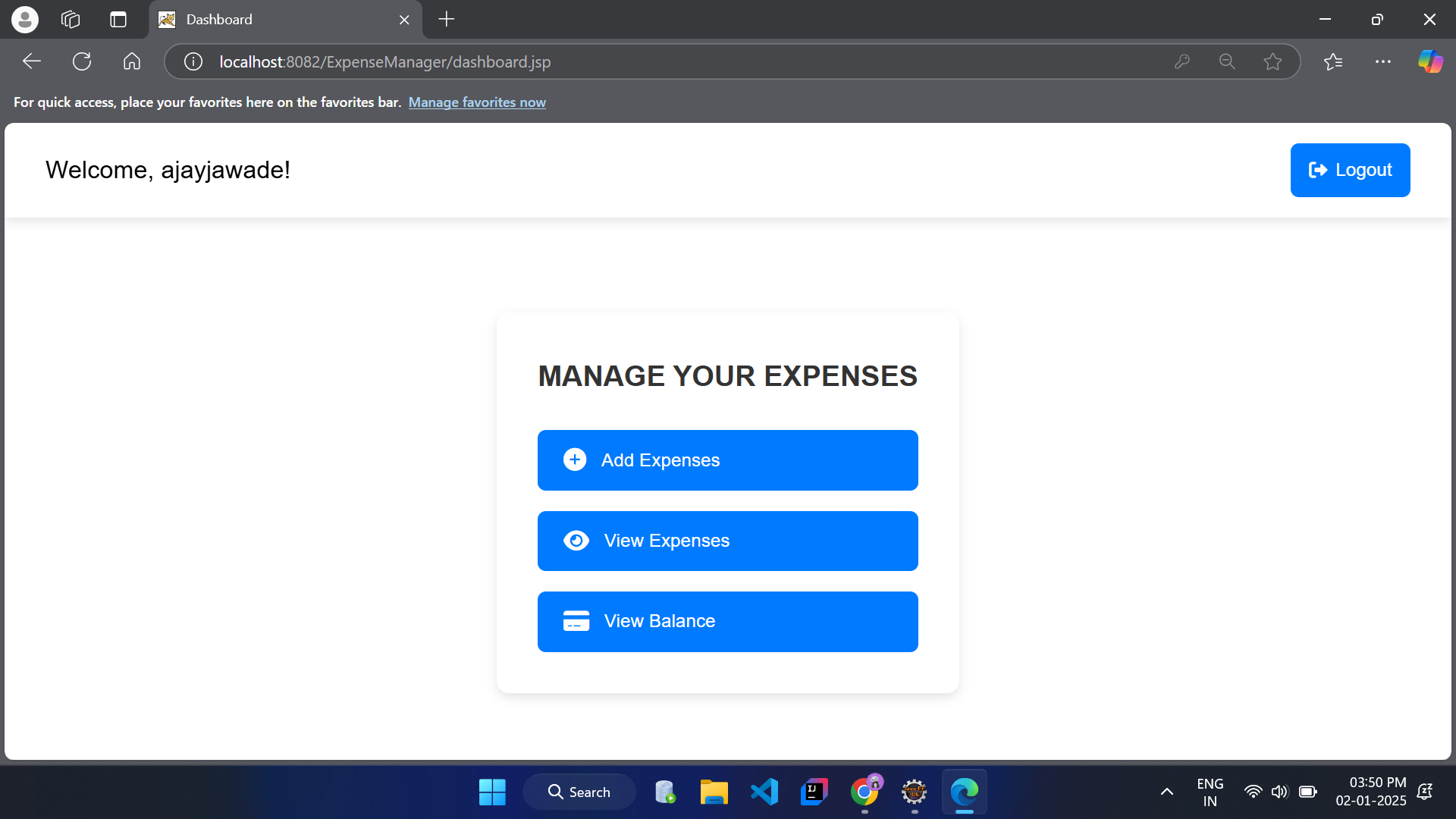
* *Java: The Complete Reference* by Herbert Schildt.
* *Database Management Systems* by Raghu Ramakrishnan.

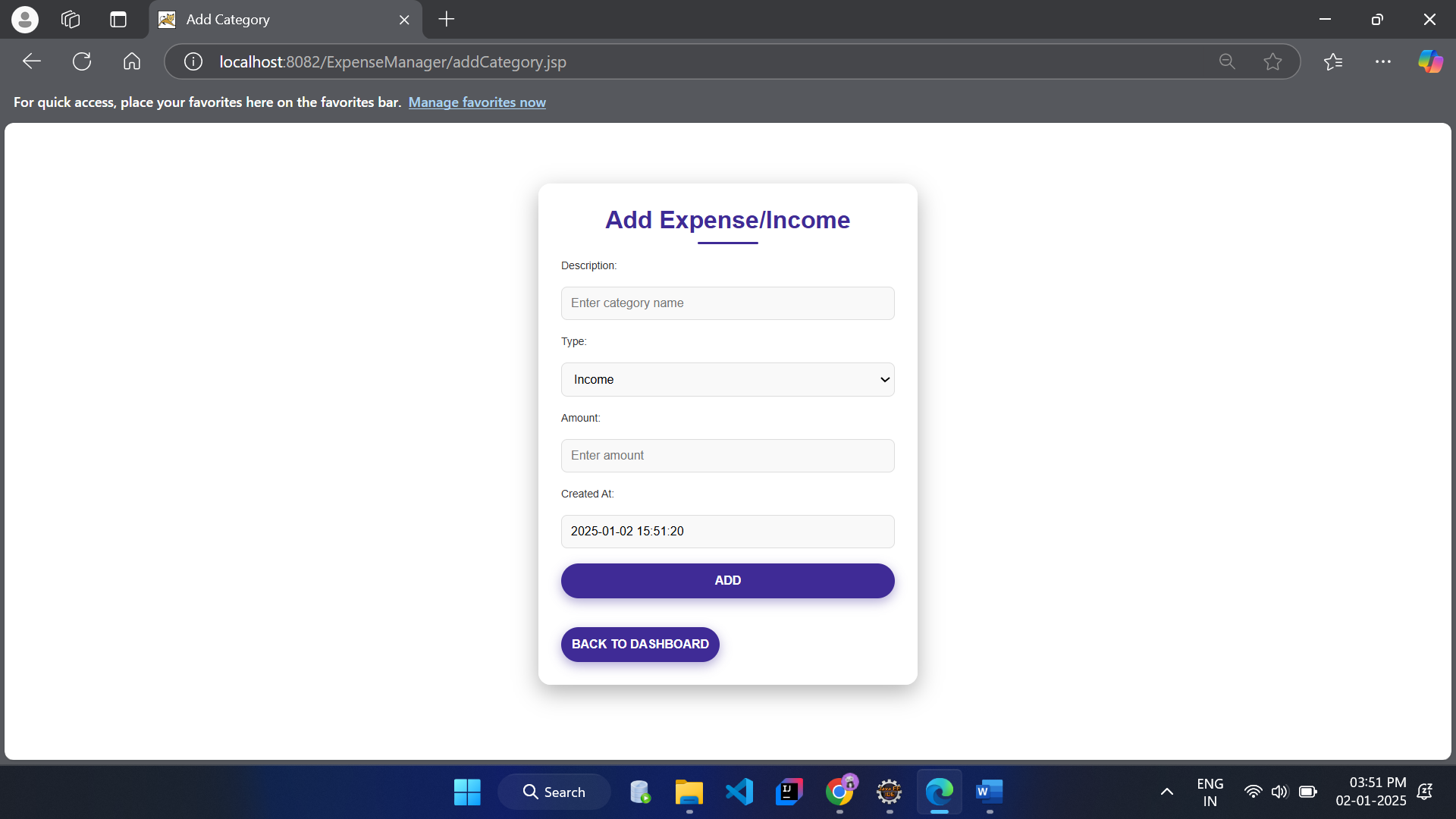
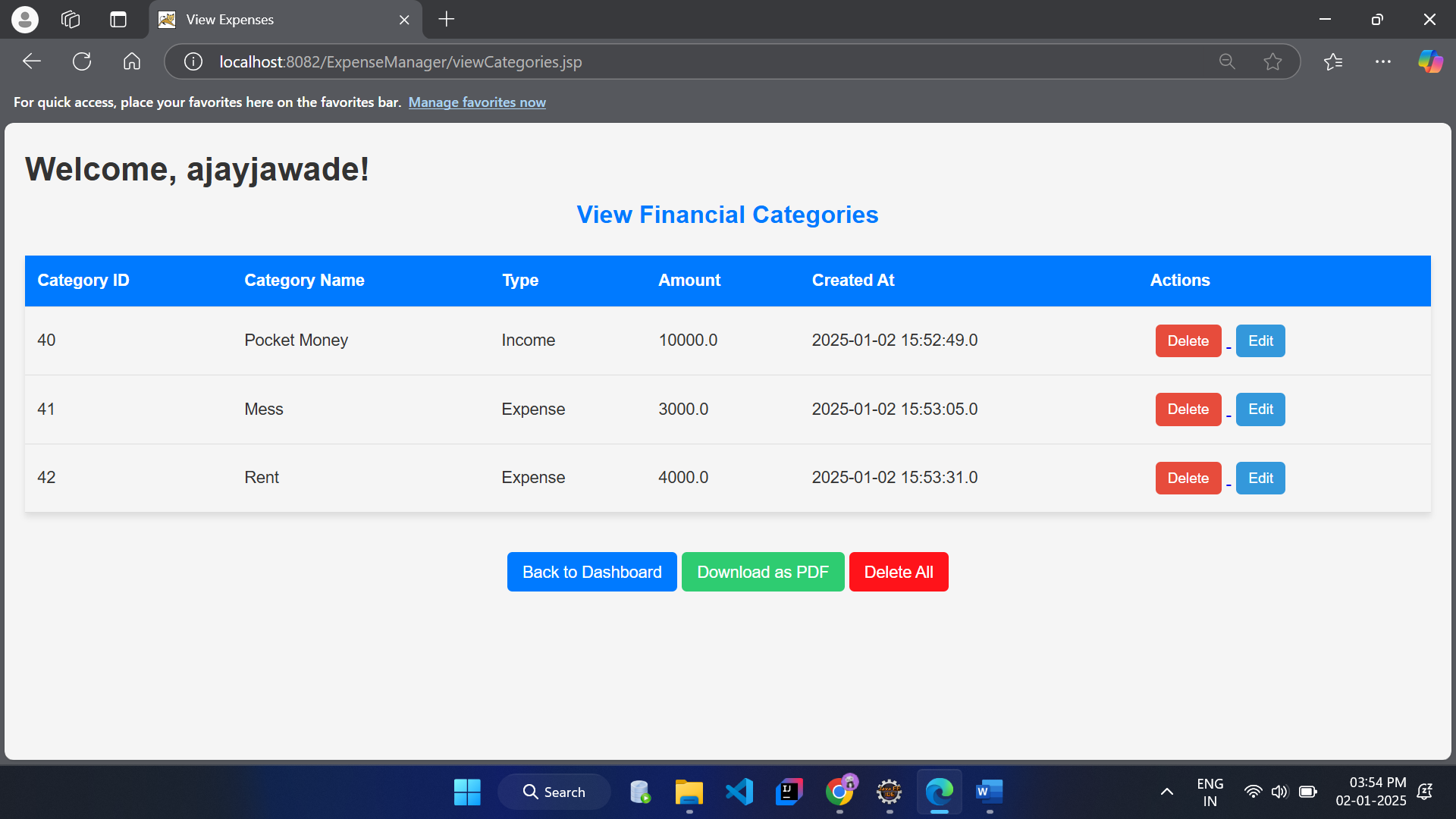
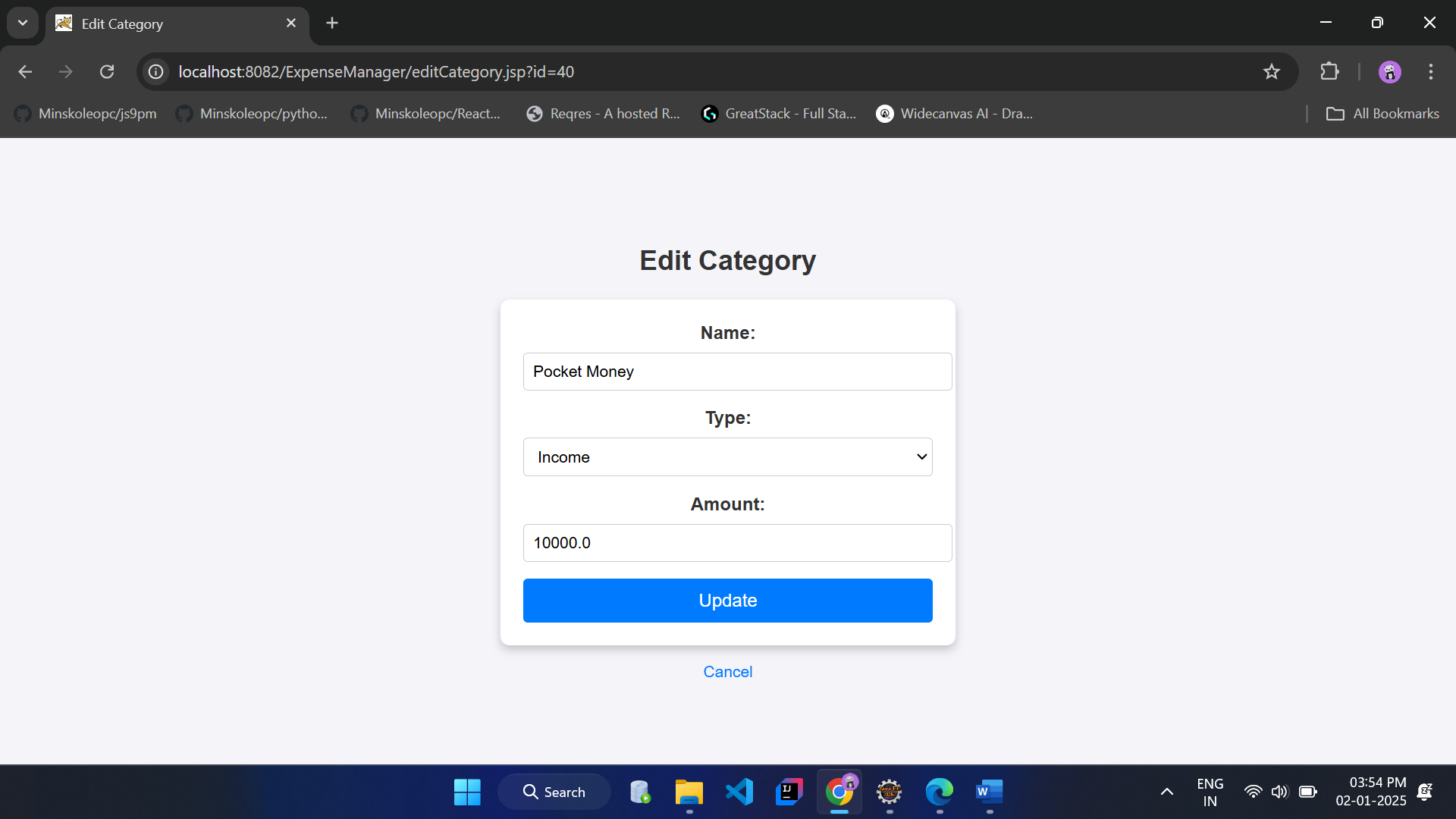
Web References

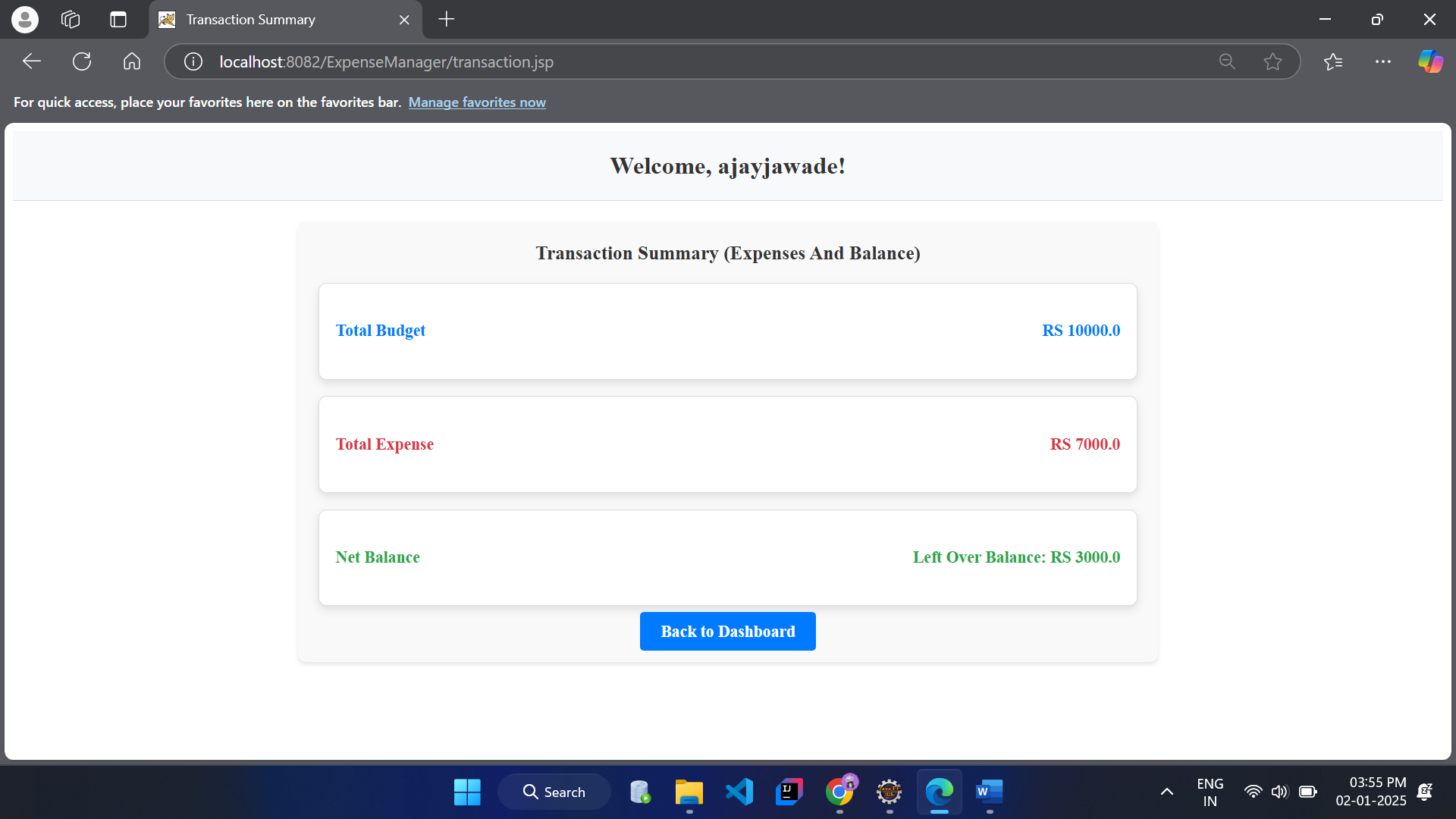
* [GitHub](https://www.github.com)
* [JavaTpoint](https://www.javatpoint.com)
* [Stack Overflow](https://www.stackoverflow.com)
* [Oracle](https://www.oracle.com)

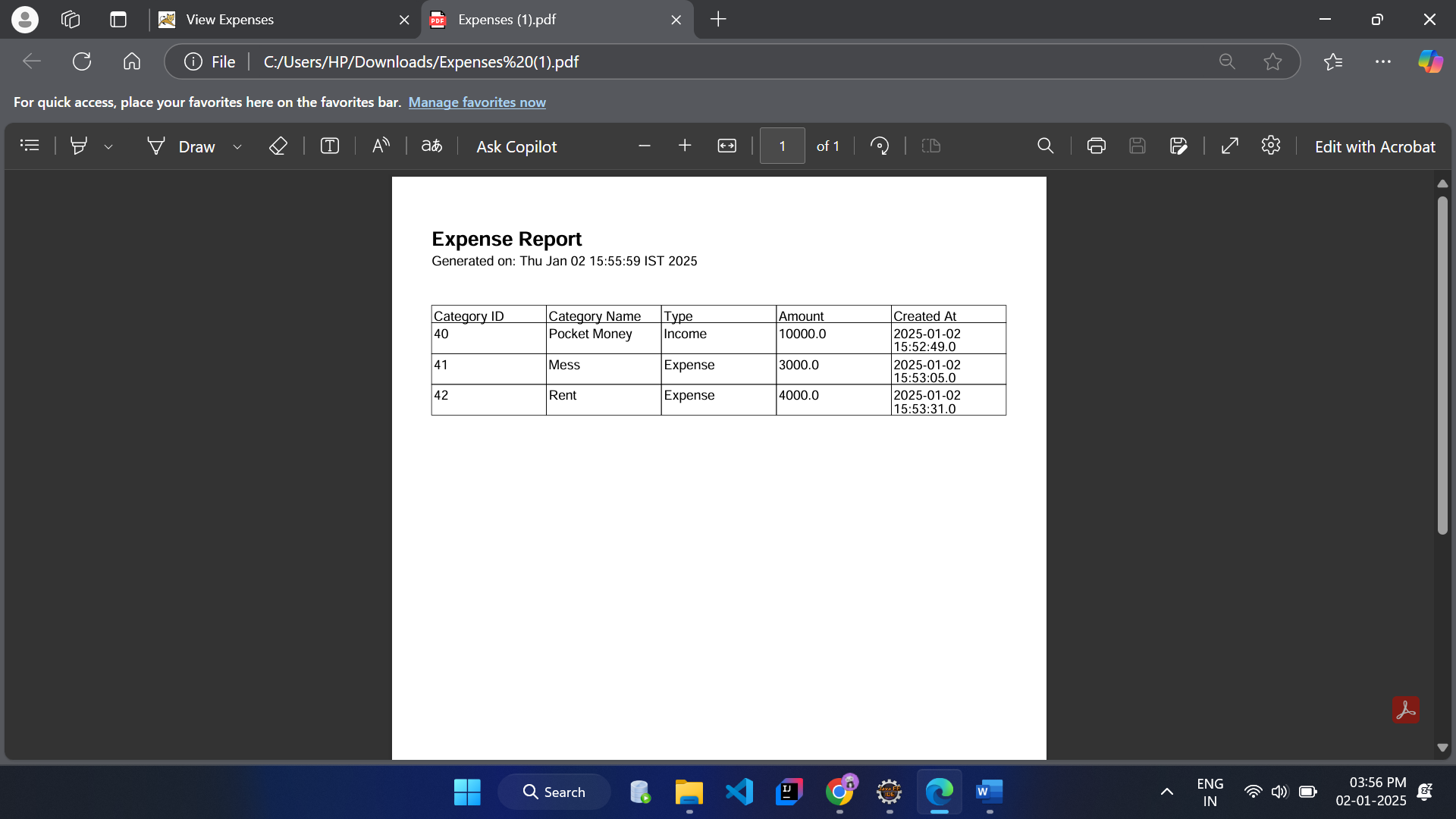
**Chapter 8: ANNEXURES**

* 1. **User Interface Screens**
  2. **Home Page**
  3. **Login Page**
  4. **Registration Page**
  5. **Dashboard Page**



* 1. **Add Category Page**
  2. **View Categories Page**
  3. **Edit Category Page**
  4. **Transaction Summary Page**



1. **PDF of Expenses And Income**
   1. **Sample Code**

**1.Login Servlet**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

@WebServlet("/loginForm")

public class Login extends HttpServlet {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/expensemanager";

private static final String DB\_USERNAME = "root";

private static final String DB\_PASSWORD = "Latari1234!";

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.setContentType("text/html");

String username = req.getParameter("username");

String password = req.getParameter("password");

// Basic empty field validation

if (username == null || username.trim().isEmpty() || password == null || password.trim().isEmpty()) {

resp.sendRedirect("login.jsp?error=Both fields are required.");

return;

}

try {

// Load database driver

Class.forName("com.mysql.cj.jdbc.Driver");

try (Connection con = DriverManager.getConnection(DB\_URL, DB\_USERNAME, DB\_PASSWORD)) {

String query = "SELECT \* FROM users WHERE username = ? AND password = ?";

try (PreparedStatement ps = con.prepareStatement(query)) {

ps.setString(1, username);

ps.setString(2, password);

try (ResultSet rs = ps.executeQuery()) {

if (rs.next()) {

// Login successful - set session attributes

HttpSession session = req.getSession();

session.setAttribute("username", rs.getString("username"));

session.setAttribute("email", rs.getString("email"));

session.setAttribute("name", rs.getString("name"));

session.setAttribute("userId", rs.getInt("id"));

// Redirect to categories/dashboard page

resp.sendRedirect("dashboard.jsp");

} else {

// Invalid credentials

resp.sendRedirect("login.jsp?error=Invalid Username or Password.");

}

}

}

}

} catch (Exception e) {

e.printStackTrace();

resp.sendRedirect("login.jsp?error=Internal server error. Please try again later.");

}

}

}

1. **Login JSP Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login - Expense Manager</title>

<link rel="stylesheet" href="css/loginstyles.css">

</head>

<body>

<div class="login-wrapper">

<div class="login-container">

<h1 class="login-header">Welcome Please Login</h1>

<!-- Error message display -->

<%

String errorMessage = request.getParameter("error");

if (errorMessage != null) {

%>

<p class="error-message"><%= errorMessage %></p>

<%

}

%>

<form action="loginForm" method="post" class="login-form">

<div class="input-group">

<label for="username">Username</label>

<input type="text" id="username" name="username" placeholder="Enter your username" required>

</div>

<div class="input-group">

<label for="password">Password</label>

<input type="password" id="password" name="password" placeholder="Enter your password" required>

</div>

<input type="submit" value="Login" class="btn-login">

</form>

<div class="register-link">

<p>Don't have an account? <a href="register.jsp">Register here</a></p>

</div>

</div>

</div>

</body>

</html>

1. **Registration Servlet**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.Part;

@WebServlet("/regForm")

public class Register extends HttpServlet {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/expensemanager";

private static final String DB\_USERNAME = "root";

private static final String DB\_PASSWORD = "Latari1234!";

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.setContentType("text/html");

PrintWriter out = resp.getWriter();

String name = req.getParameter("name");

String email = req.getParameter("email");

String username = req.getParameter("username");

String password = req.getParameter("password");

try {

// Load MySQL Driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Establish connection

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USERNAME, DB\_PASSWORD)) {

// Check if the username already exists

String checkUserQuery = "SELECT \* FROM users WHERE username = ?";

try (PreparedStatement psCheck = conn.prepareStatement(checkUserQuery)) {

psCheck.setString(1, username);

ResultSet rs = psCheck.executeQuery();

if (rs.next()) {

out.print("<h3 style='color:red;'>Username already exists. Please choose another.</h3>");

RequestDispatcher rd = req.getRequestDispatcher("/register.jsp");

rd.include(req, resp);

return;

}

}

// Insert user details into the database

String insertQuery = "INSERT INTO users (name, username, email, password) VALUES (?, ?, ?, ?)";

try (PreparedStatement psInsert = conn.prepareStatement(insertQuery)) {

psInsert.setString(1, name);

psInsert.setString(2, username);

psInsert.setString(3, email);

psInsert.setString(4, password);

int result = psInsert.executeUpdate();

if (result > 0) {

out.print("<h3 style='color:green;'>User Registered Successfully!</h3>");

resp.sendRedirect("login.jsp");

} else {

out.print("<h3 style='color:red;'>User could not be registered. Please try again.</h3>");

RequestDispatcher rd = req.getRequestDispatcher("/register.jsp");

rd.include(req, resp);

}

}

}

} catch (Exception e) {

e.printStackTrace();

out.print("<h3 style='color:red;'>An error occurred: " + e.getMessage() + "</h3>");

RequestDispatcher rd = req.getRequestDispatcher("/register.jsp");

rd.include(req, resp);

}

}

}

1. **Dashboard JSP**

<%@ page import=*"java.util.\*"* %>

<%

// Ensure only logged-in users can access this page

String username = (String) session.getAttribute("username");

if (username == null) {

response.sendRedirect("login.jsp?error=Please login to continue.");

}

%>

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<title>Dashboard</title>

<link rel=*"stylesheet"* href=*"css/dashboardstyles.css"*>

<link rel=*"stylesheet"* href=*"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"*> <!-- Font Awesome -->

</head>

<body>

<header>

<nav class=*"navbar"*>

<h2 class=*"welcome-text"*>Welcome, <%= username %>!</h2>

<div class=*"spacer"*></div> <!-- Spacer to push logout button to the right -->

<a href=*"logout"* class=*"logout-btn"*><i class=*"fas fa-sign-out-alt"*></i> Logout</a>

</nav>

</header>

<!-- Main Dashboard Section -->

<div class=*"dashboard-container"*>

<h3 class=*"dashboard-heading"*>Manage Your Expenses</h3>

<ul class=*"dashboard-menu"*>

<li>

<a href=*"addCategory.jsp"* class=*"menu-item"*>

<i class=*"fas fa-plus-circle"*></i> Add Expenses

</a>

</li>

<li>

<a href=*"viewCategories.jsp"* class=*"menu-item"*>

<i class=*"fas fa-eye"*></i> View Expenses

</a>

</li>

<li>

<a href=*"transaction.jsp"* class=*"menu-item"*>

<i class=*"fas fa-credit-card"*></i> View Balance

</a>

</li>

</ul>

</div>

</body>

</html>