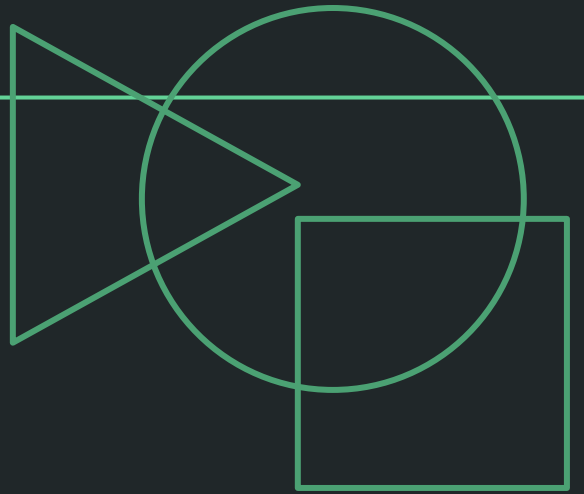
Dot Net Full Stack Training Project

LOAN MANAGEMENT SYSTEM



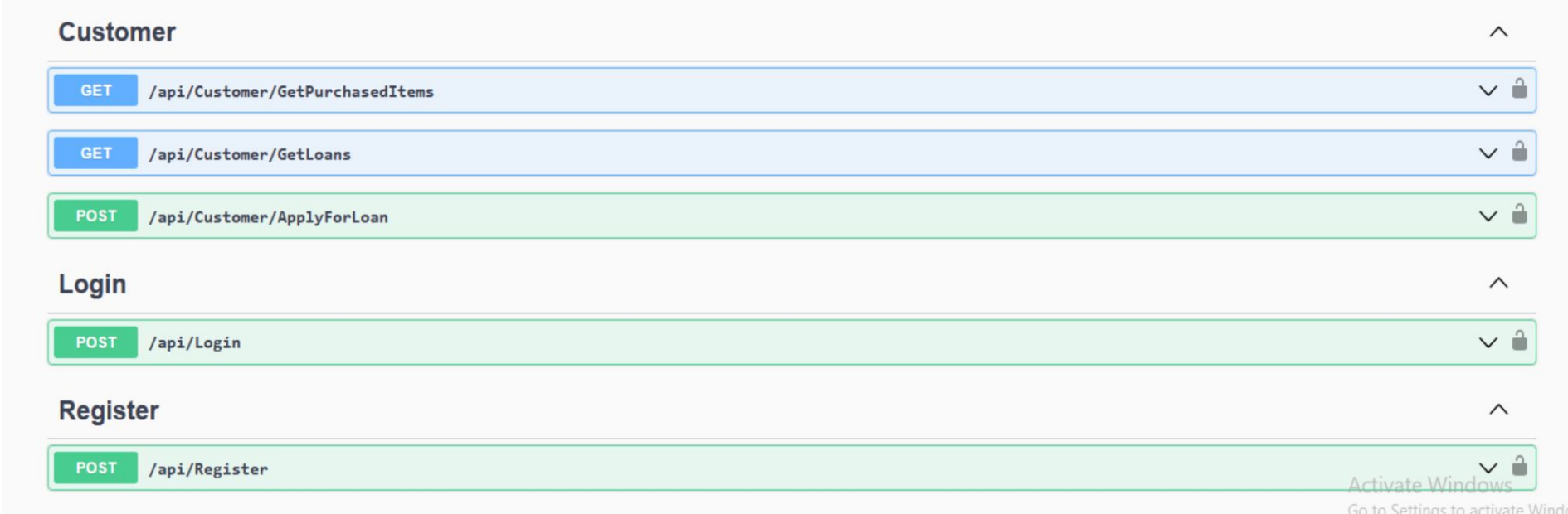
Presented by: Batch 9 (Team 3)
Instructor: Shrivalli Maheshwaran

Contributions

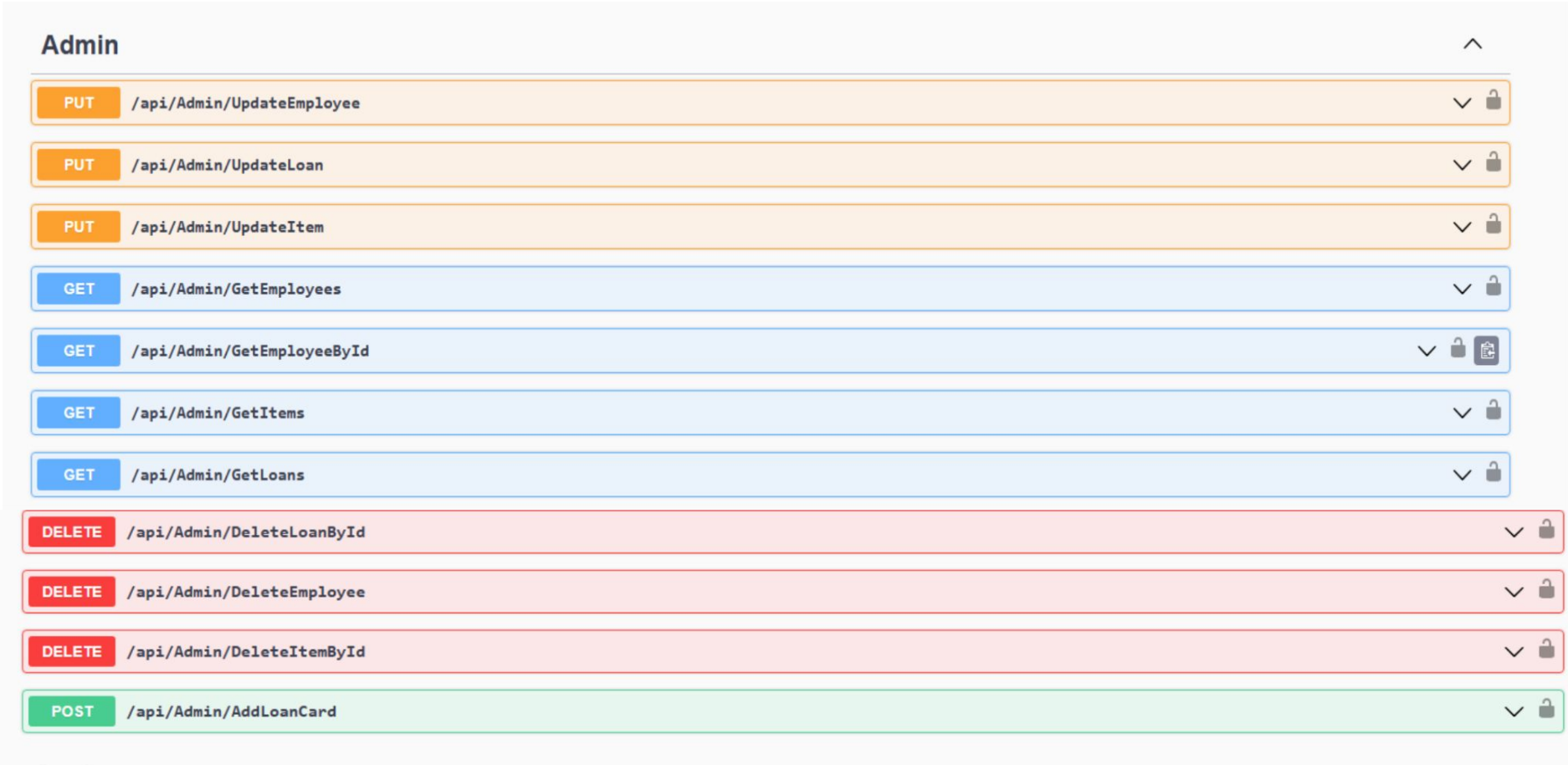
- **Ashwin Menon** - Frontend Alerts, Exception Handling, (UI - Customer management, Item Purchased, View Loans, Navbar, Dashboard) & (APIs - Edit/Delete Employee, Get Purchased Items, Get Loan Cards, Role Based Auth), Test cases for GetLoanCards API.
- **Agastya Varma** - Frontend Login/Register page, Setting up AuthContext, Password hashing, APIs - Delete loan, Login, Register, Database Schema, DDL commands.
- **Soumil Kamat** - (UI- Apply for Loan, Edit loan, Add item, Bootstrap setup, UI for Tables, adding session storage), (APIs- Delete Item, Add loan card), (API connection from frontend - delete item, delete loan)
- **Hardik Sharma**- Frontend: Reusable TableComponent, UI enhancement of master tables and edit form, helper filter function, Backend: Unit tests for GetItems() API, APIs: IssueItem,UpdateItem,UpdateLoan, DeleteItem, exception for repos, DML commands
- **Manav Munjal** - (Ui tasks- User & Admin Dashboards, ApplyForLoan, Add Employee, Dates validation), (Backend tasks- Apis - (Login, Register, GetEmployeeById), Dates validation), DDL commands, Tables creation and population sql.

Code Structure

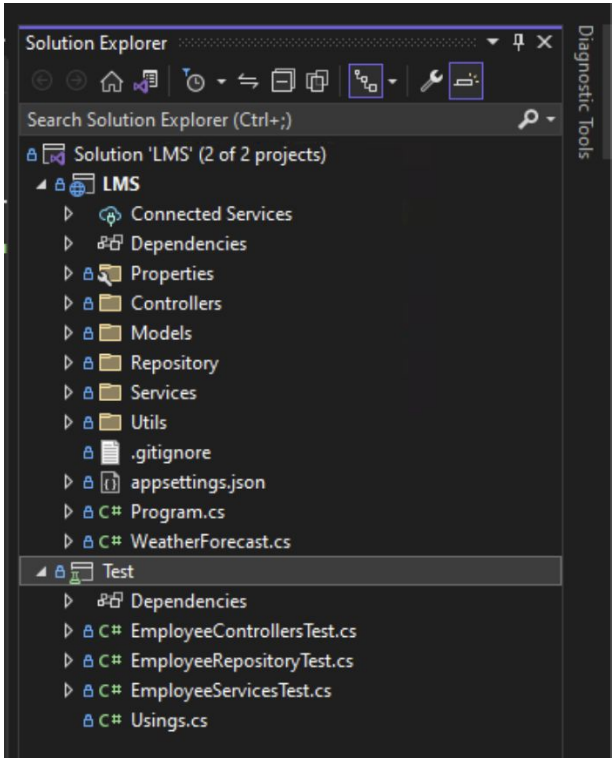
API Specification (Swagger)



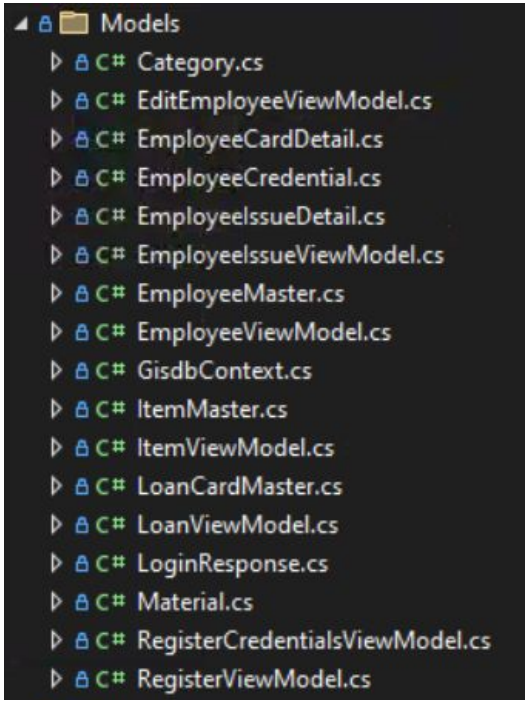
API Specification (Swagger)

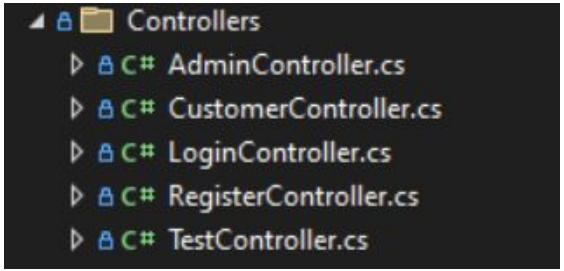


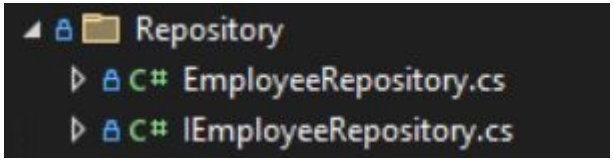
Backend Structure

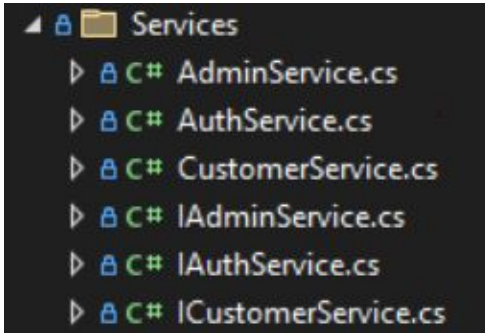


Backend Structure

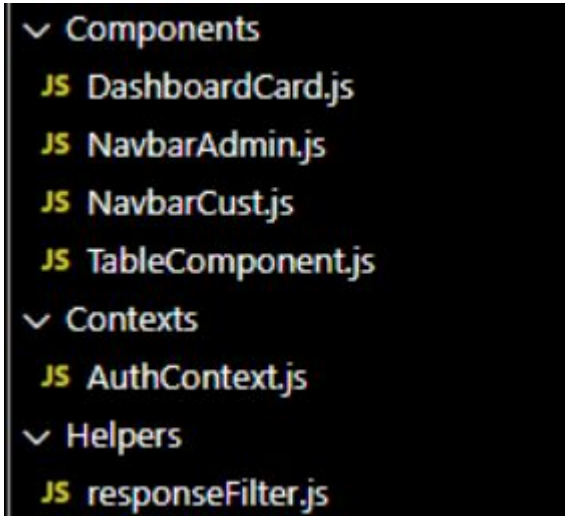


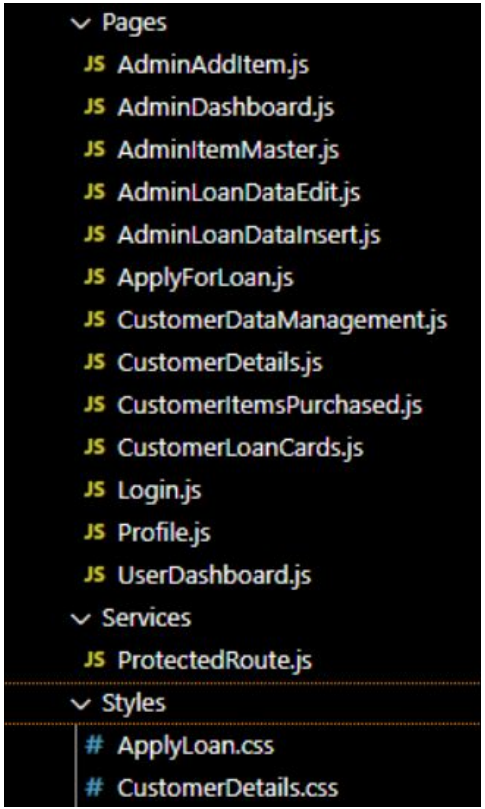




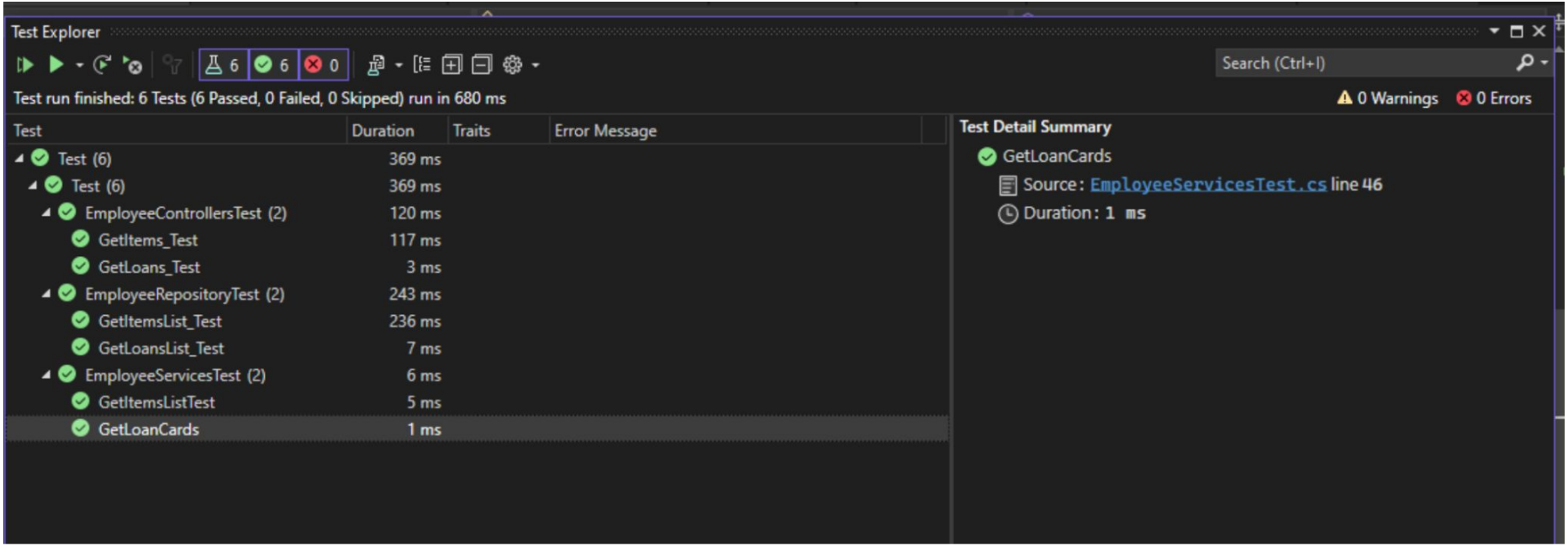


Frontend File Structure

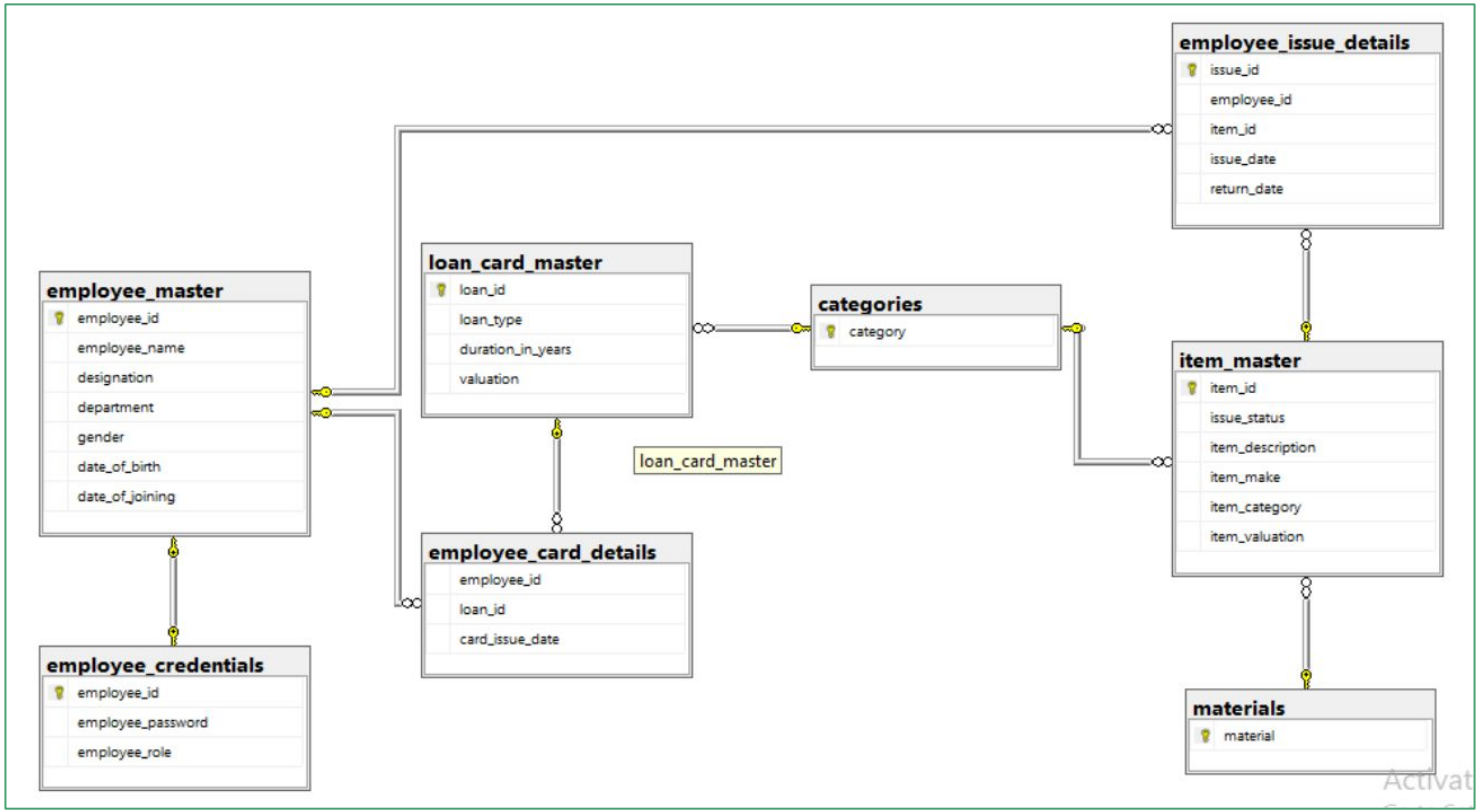




Unit Testing Results



Database Schema



Special Features

Special Features

- Appropriate error handling like validation error is shown in case employee enters date of birth > date of joining
- User is shown appropriate status, e.g. when tables do not have any data, proper message is displayed instead of blank screen
- Data consistency has been taken care of, e.g. when deleting a loan card, associated items will be deleted as well
- Navbar has been added with user greeting on right side
- Loan card valuation is displayed which is the sum of all items purchased with it
- Displaying return date for items purchased

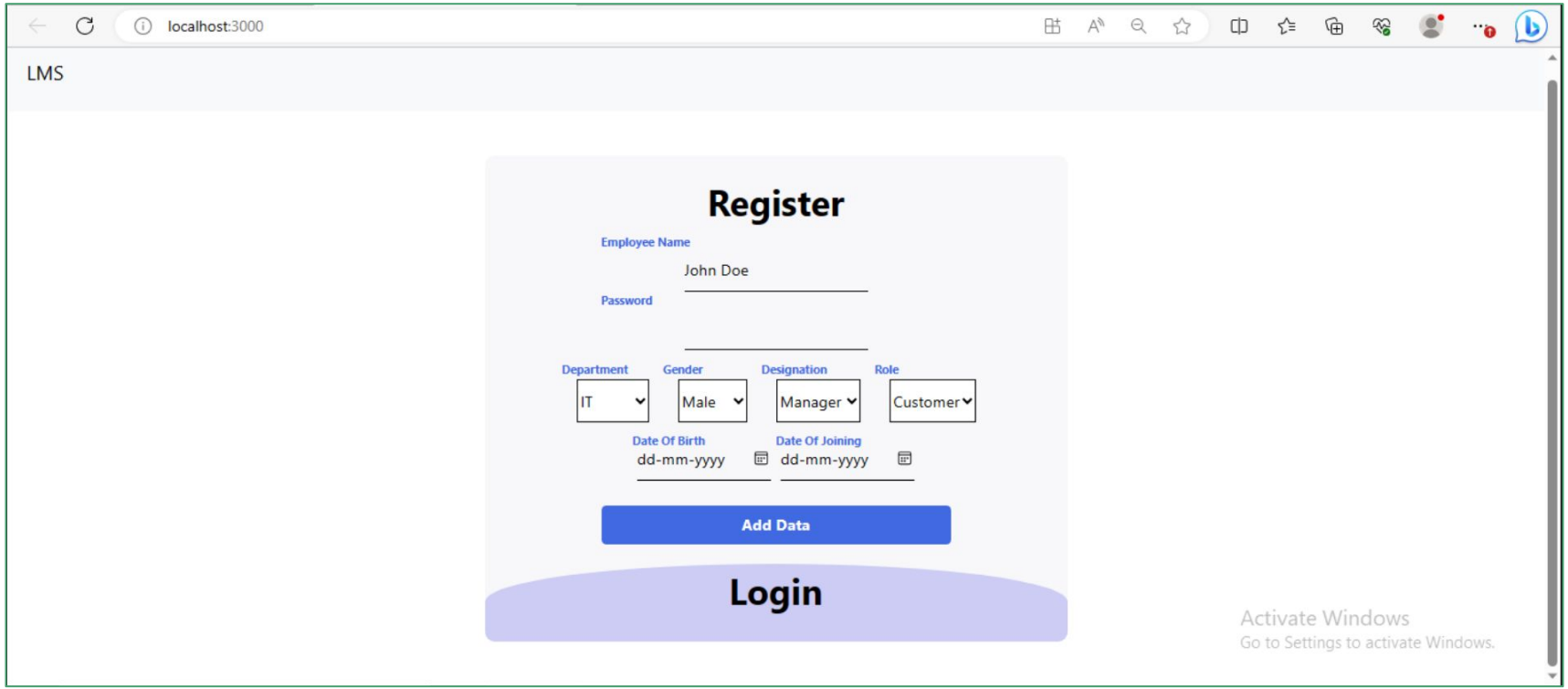
Reusable Table Component

```
<TableComponent
  headerData={["Loan ID", "Loan Type", "Duration (in years)", "Card Issue Date", "Valuation"]}
  tableData={responseFilter(loanCards, ["loanId", "loanType", "durationInYears", "cardIssueDate", "valuation"])}
  noDataMessage={{ title: "No loan cards are availed", message: "Use the Apply for Loan option to add a loan card" }} />
```

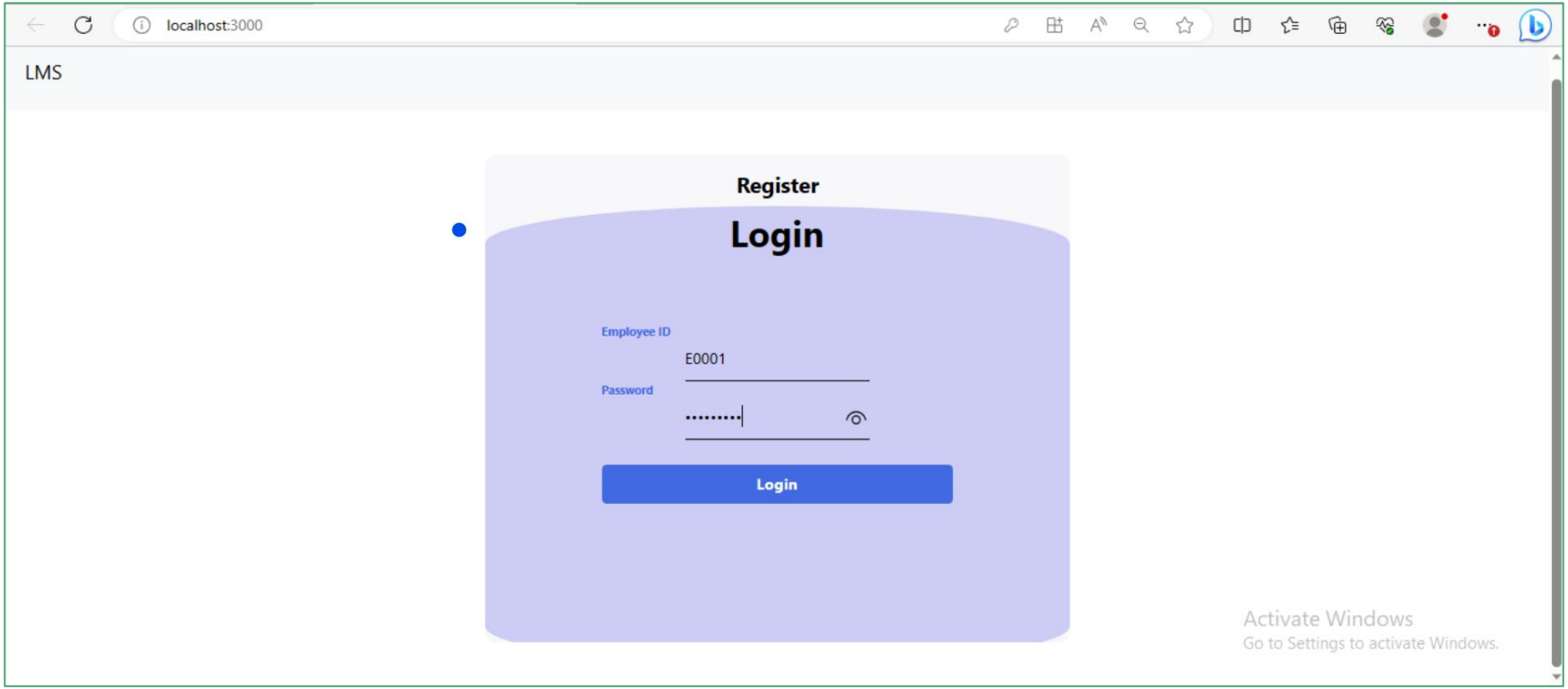
The TableComponent takes various table header names, the corresponding row data and messages to display when no data is present in the table as prop arguments. All view table pages make use of this component. It also uses a helper function responseFilter(), which filters requisite columns from backend response.

Application Walkthrough

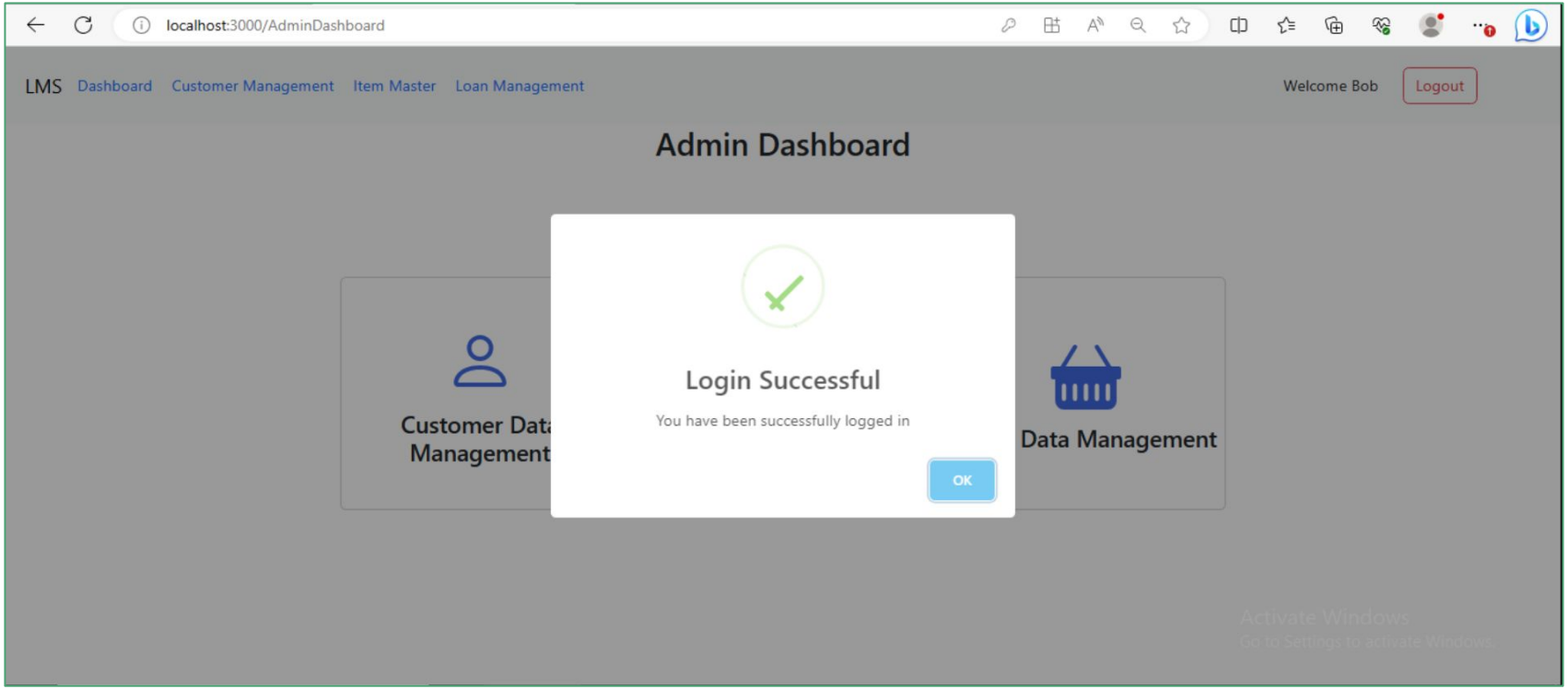
Register Screen



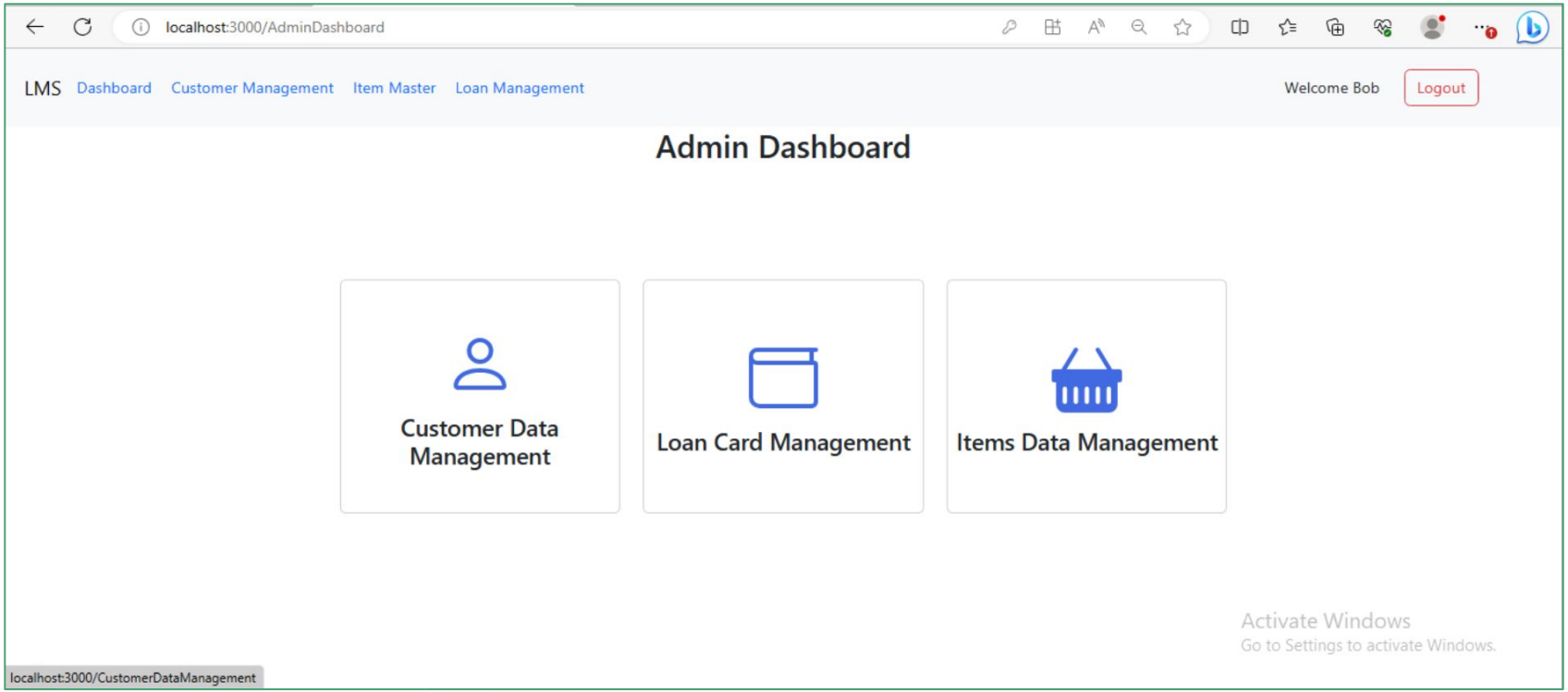
Login Screen



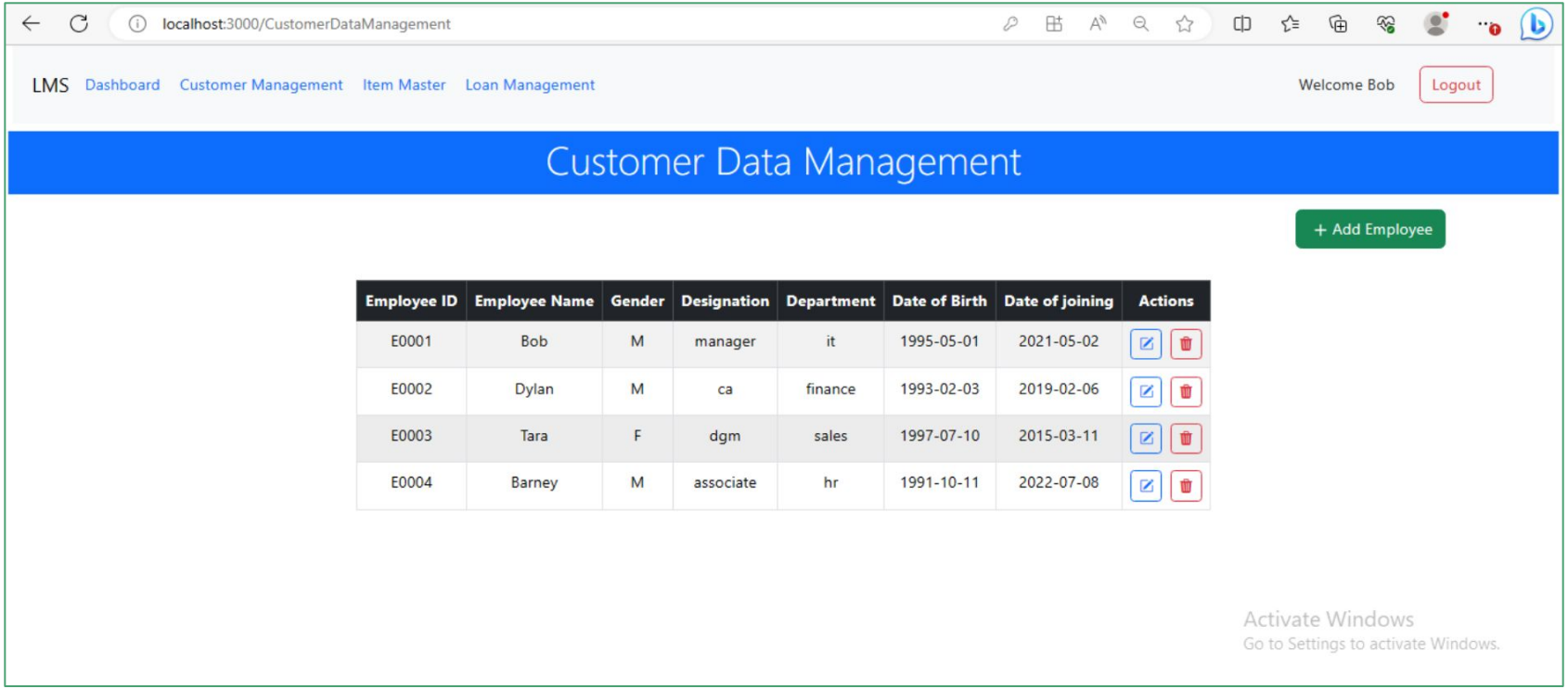
Successful Login



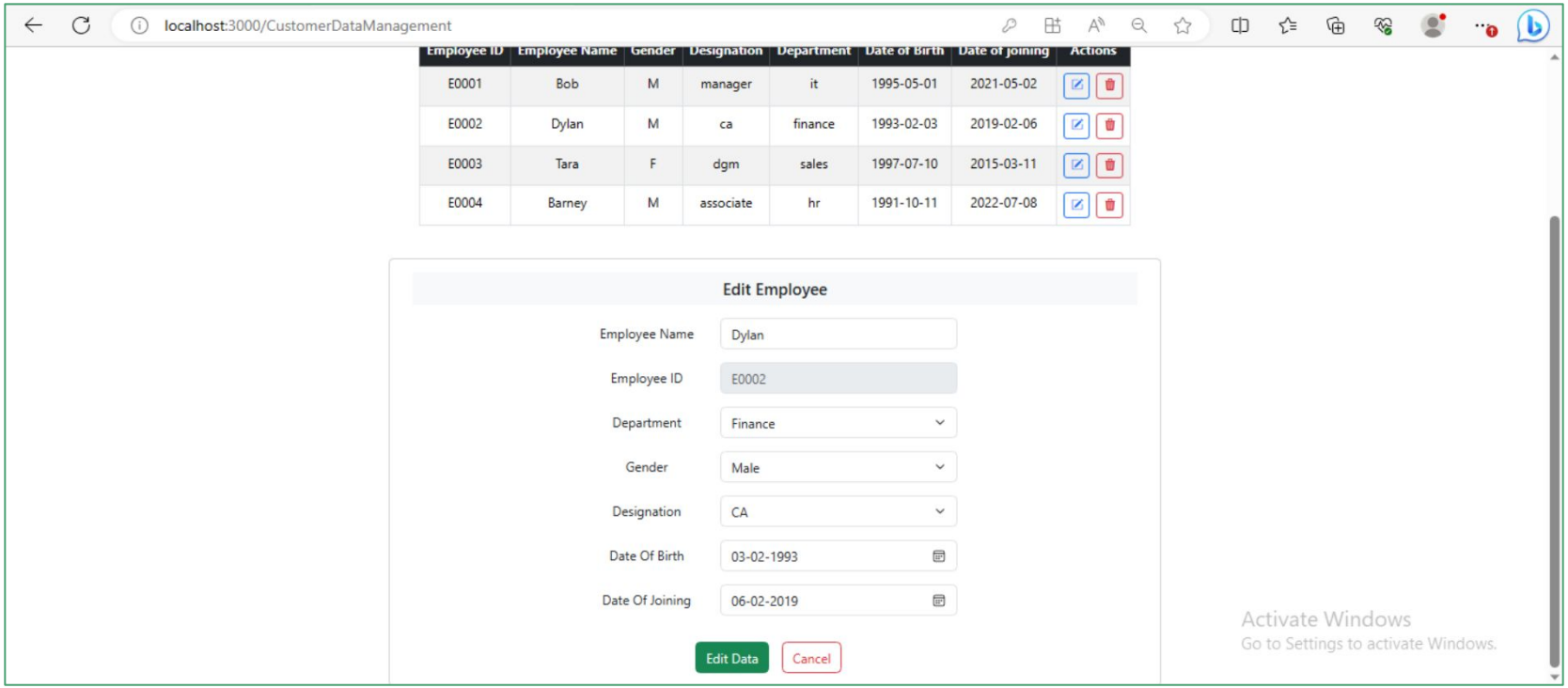
Admin Dashboard



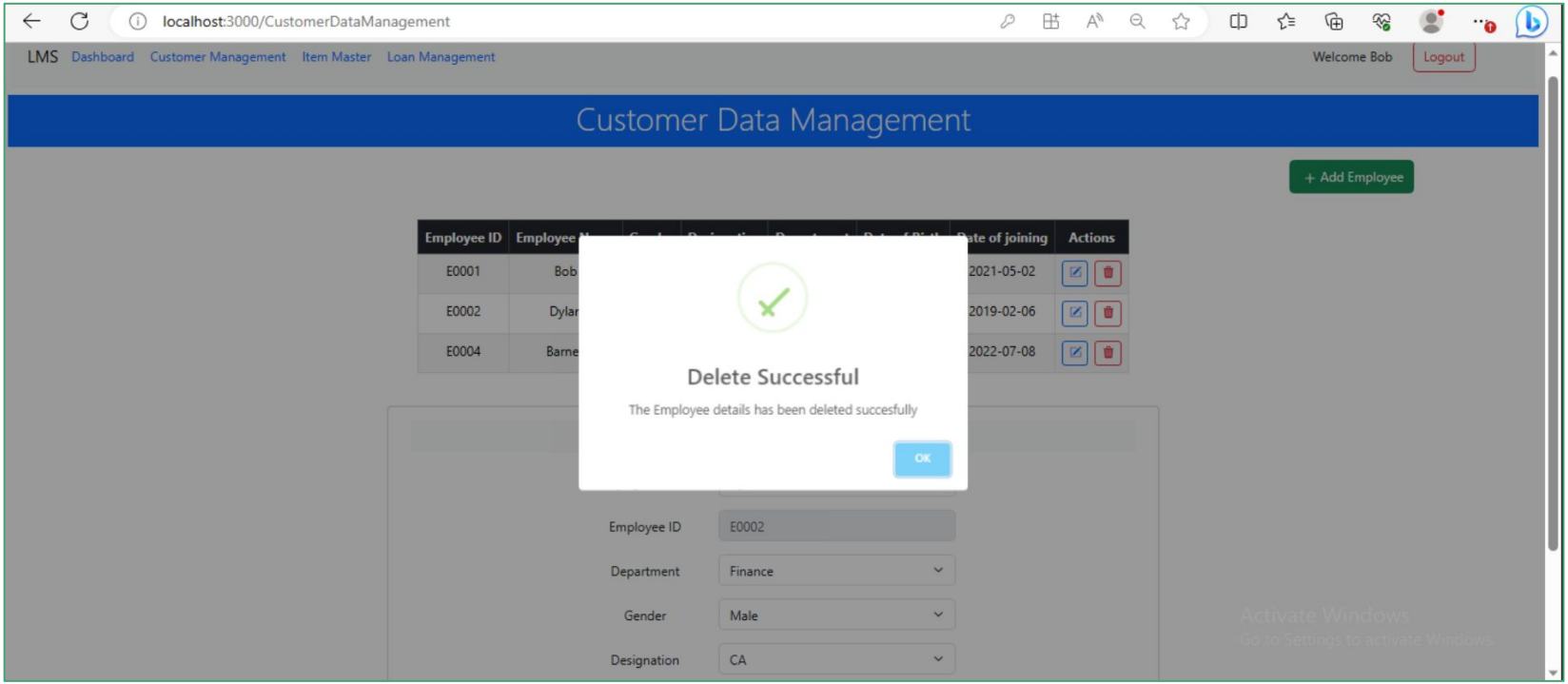
Admin > Customer Data Management



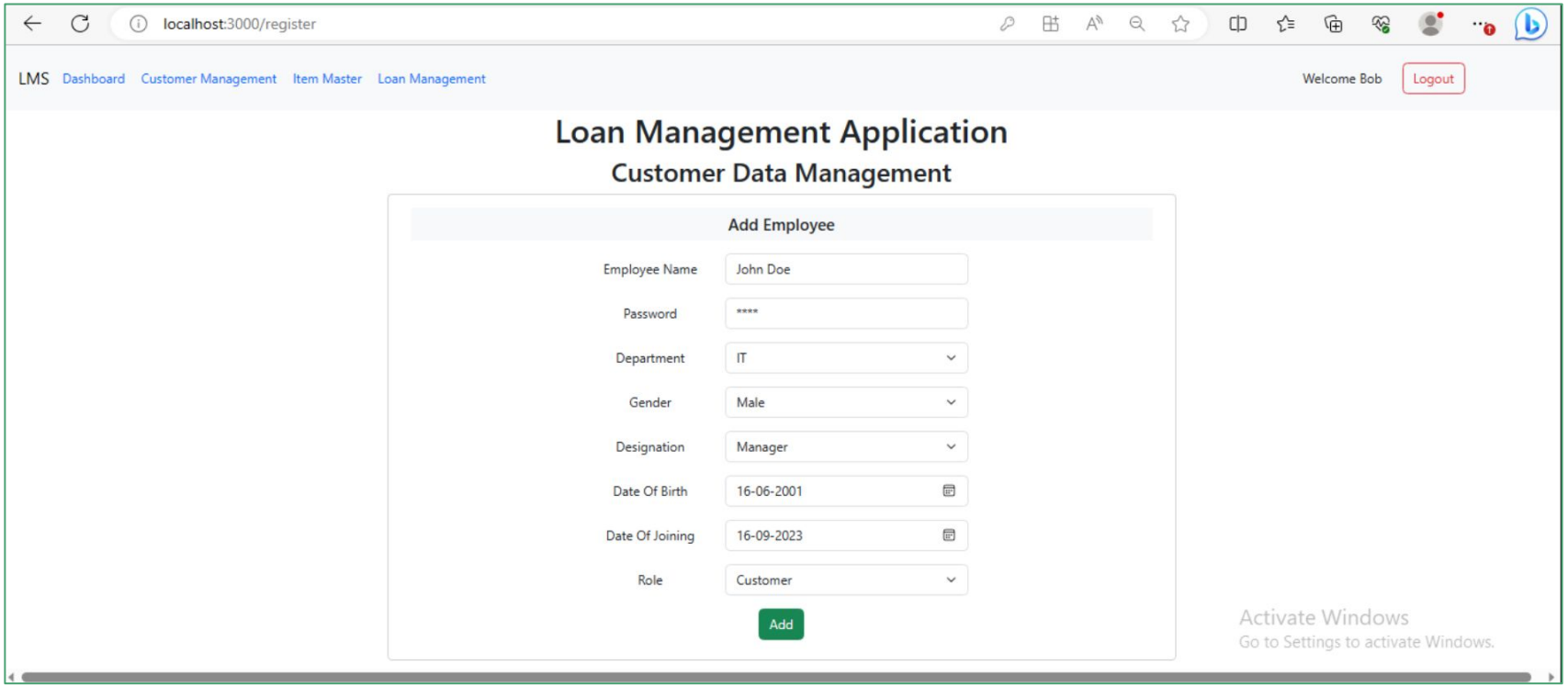
Admin > Customer Data Management > Edit Customer



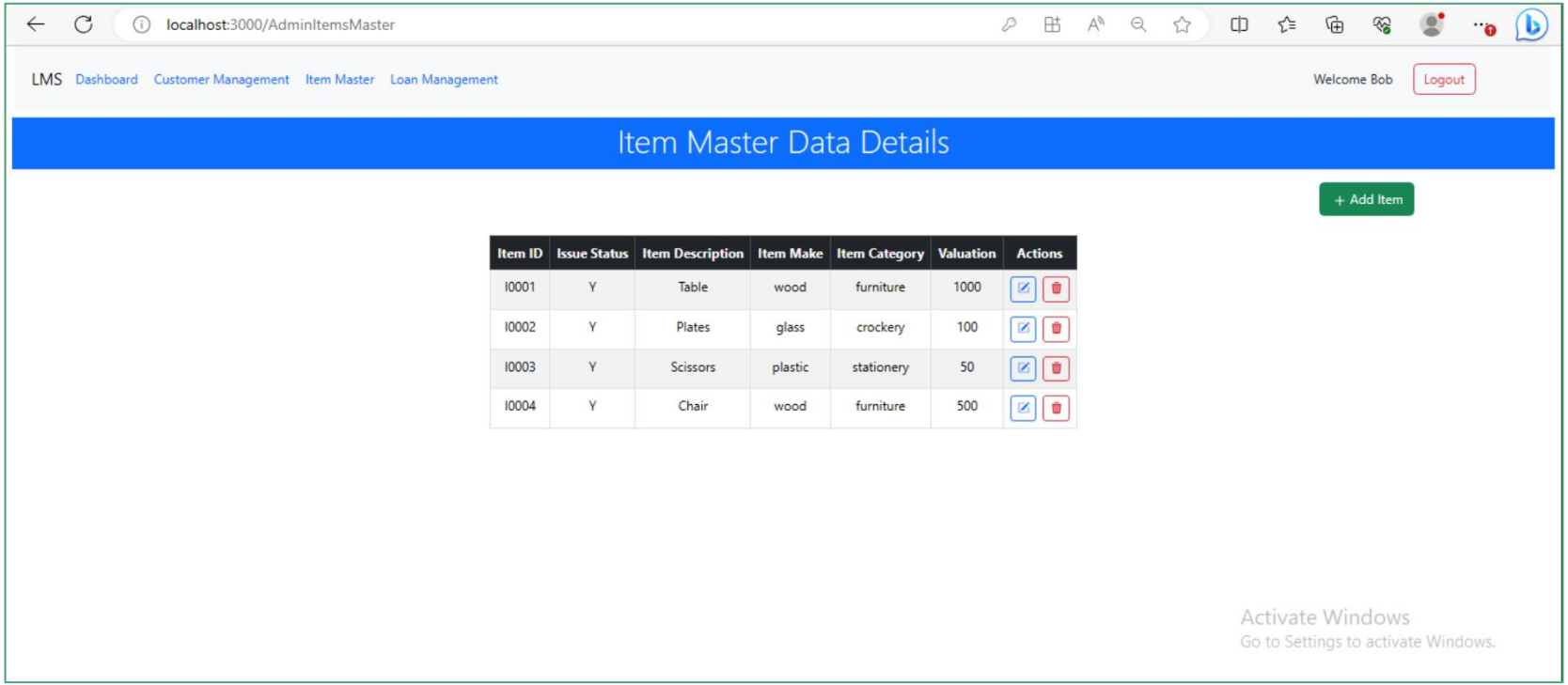
Admin > Customer Data Management > Delete Customer



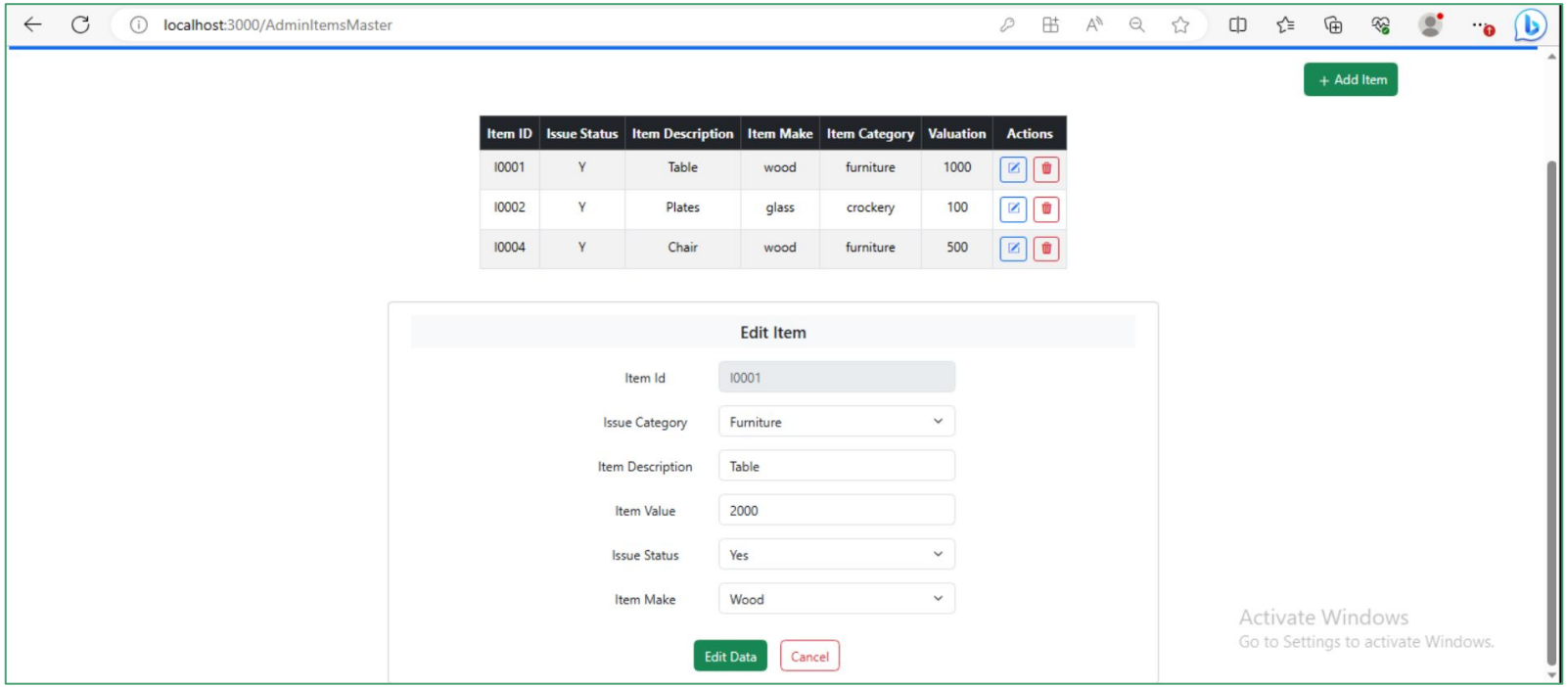
Admin > Customer Data Management > Add Employee



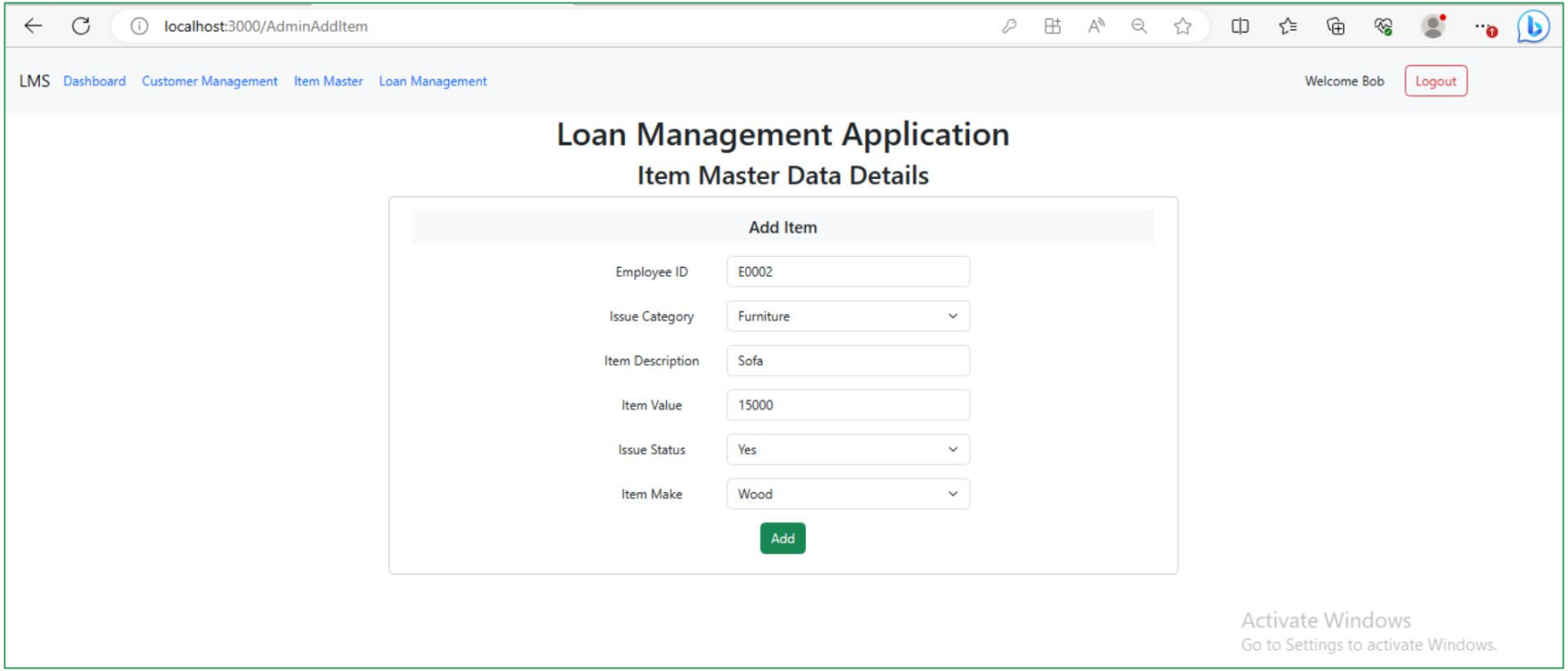
Admin > Item Master View



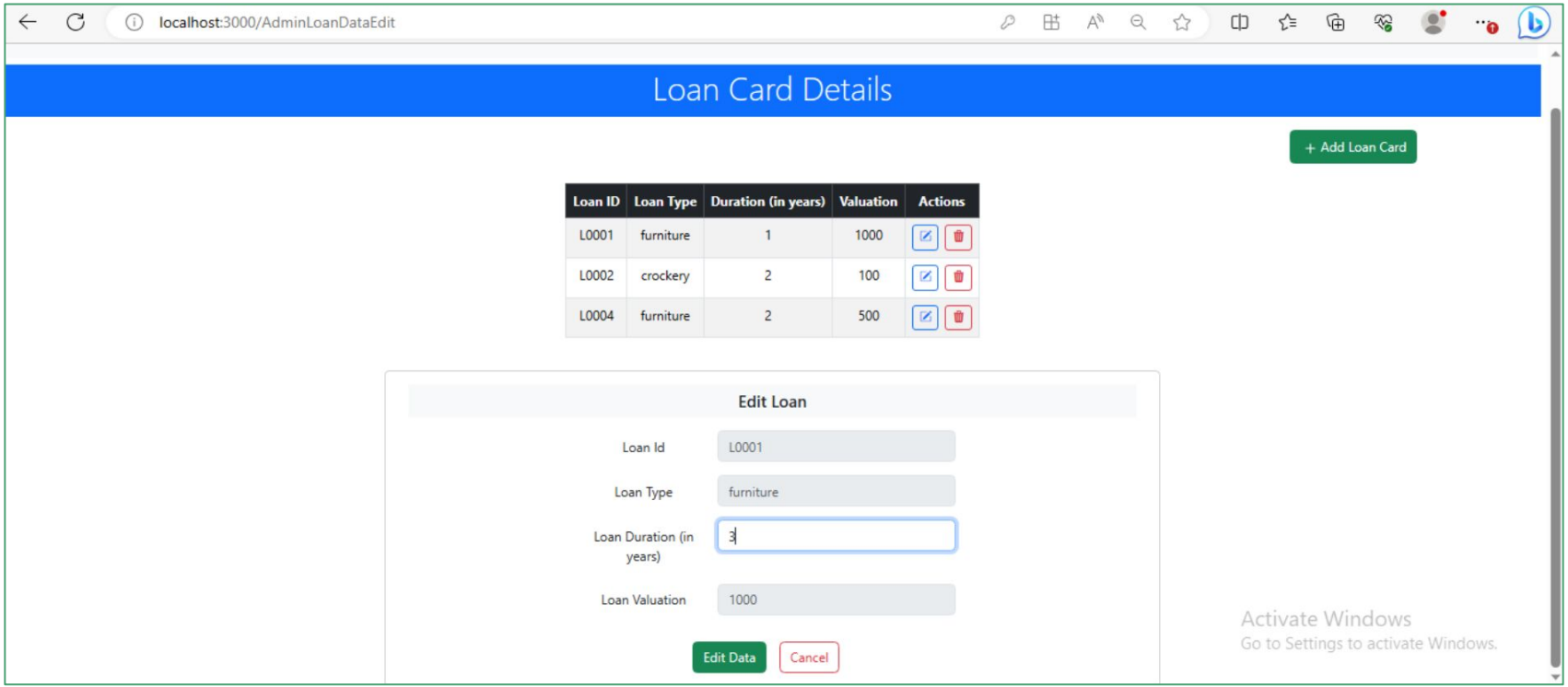
Admin > Item Master View > Edit Item



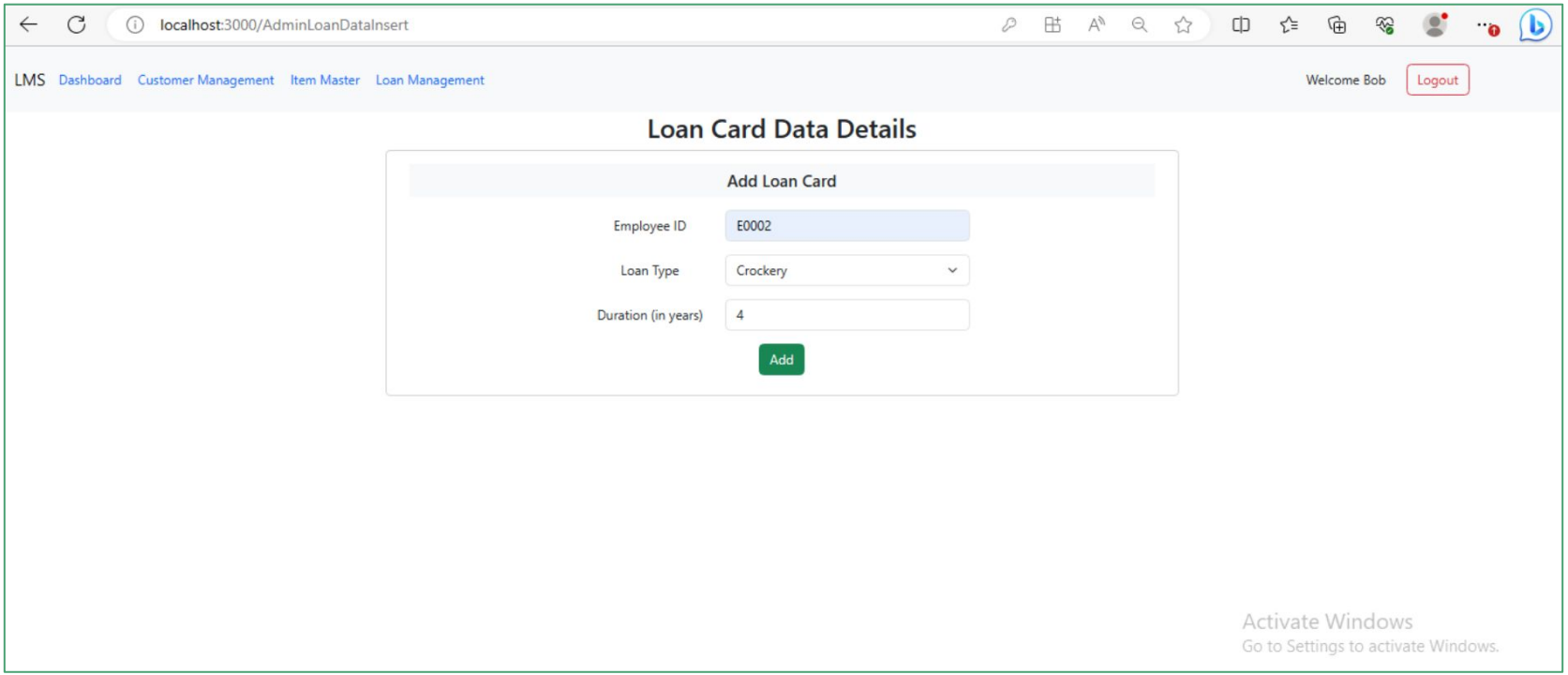
Admin > Add Item Form



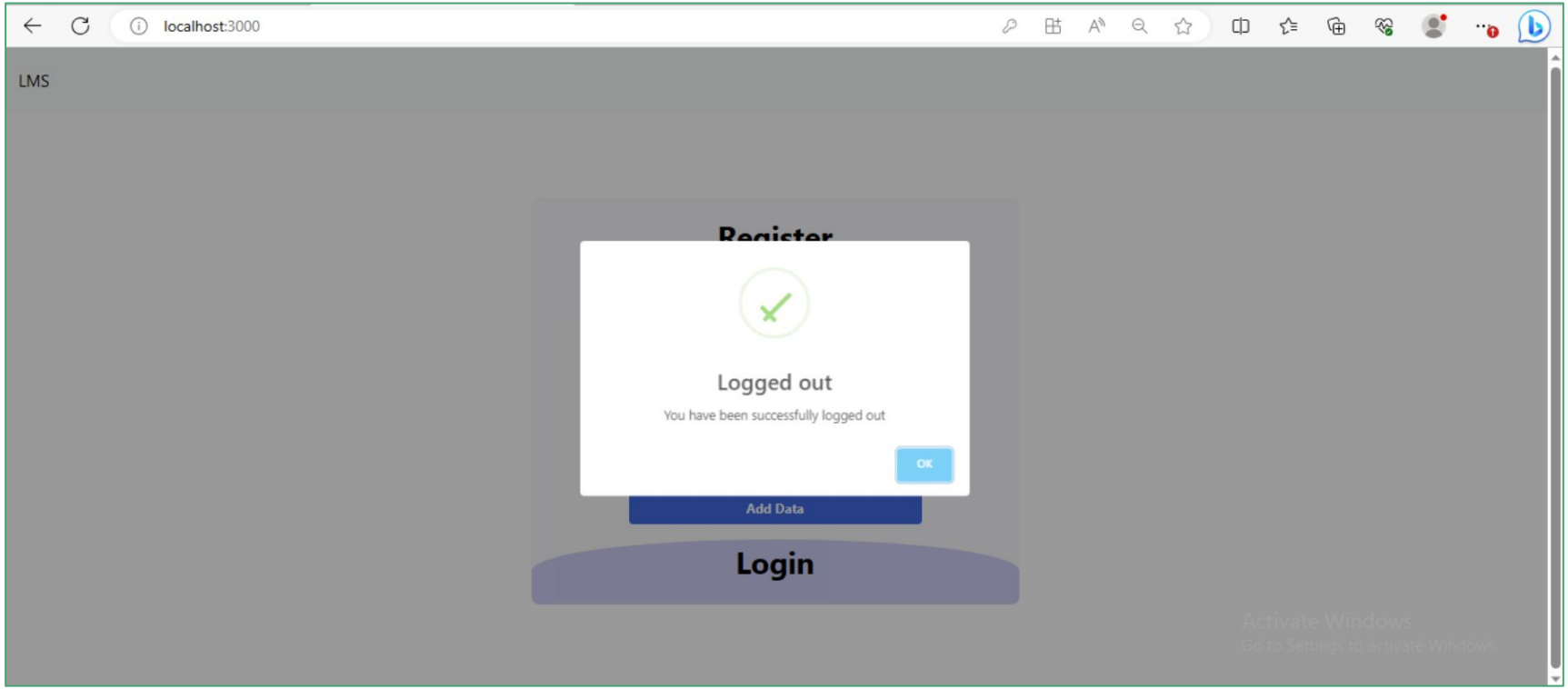
Admin > Loan Card Master and Edit Loan



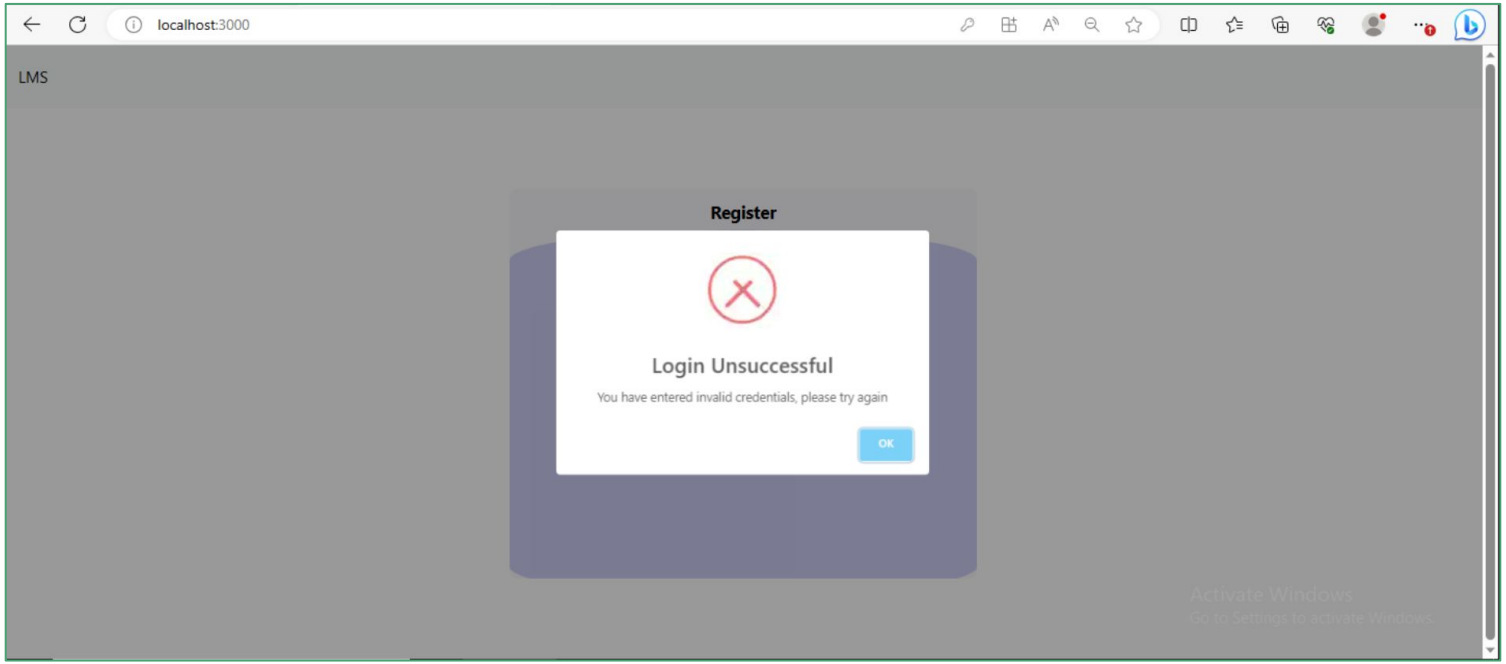
Admin > Add Loan Card



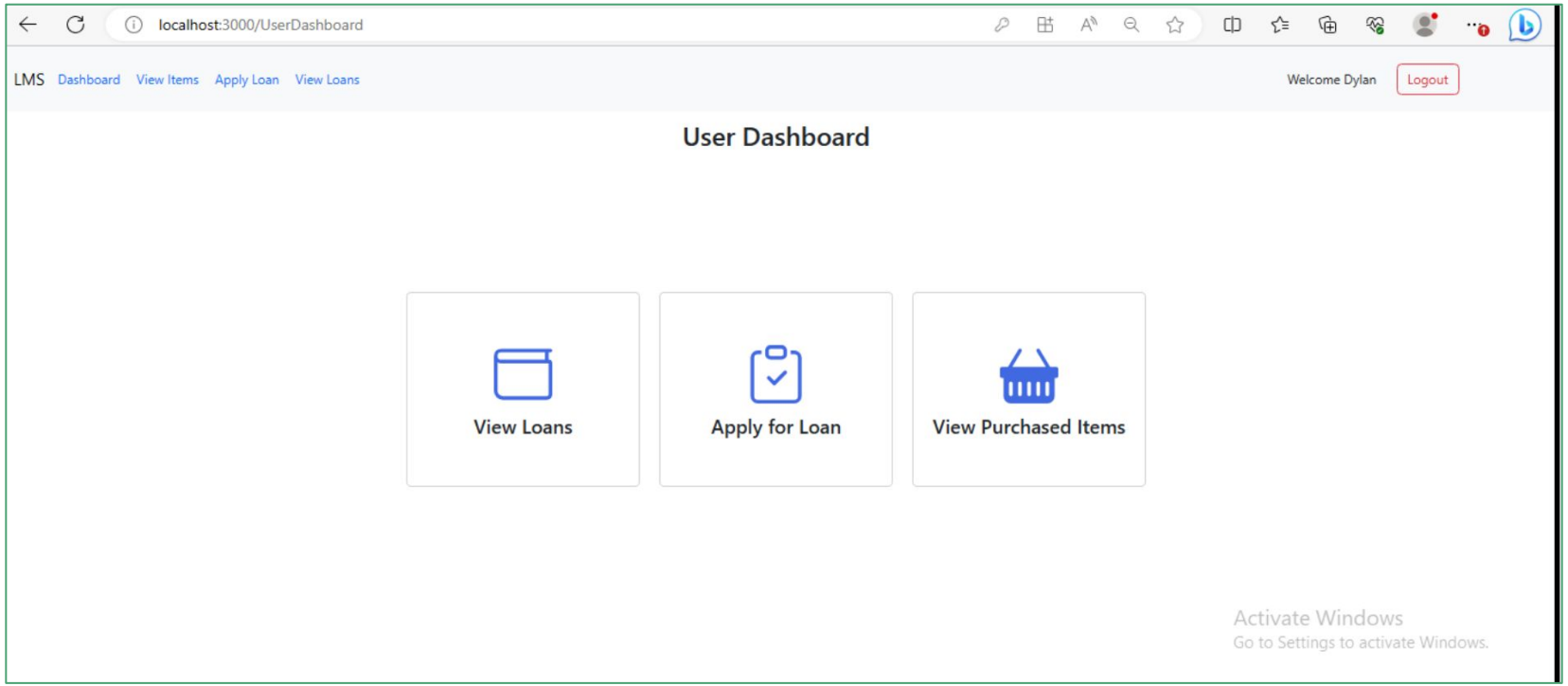
Logout Screen



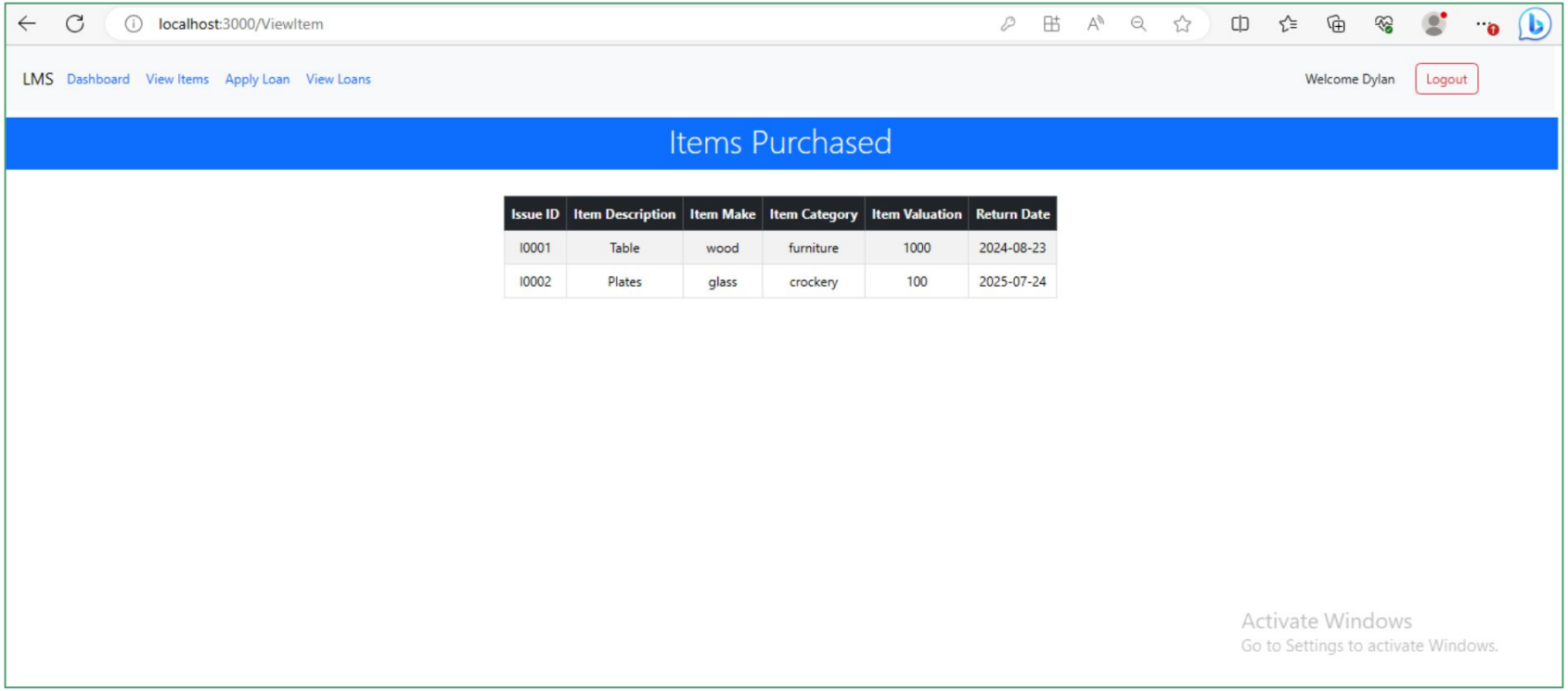
Error screen on entering invalid credentials



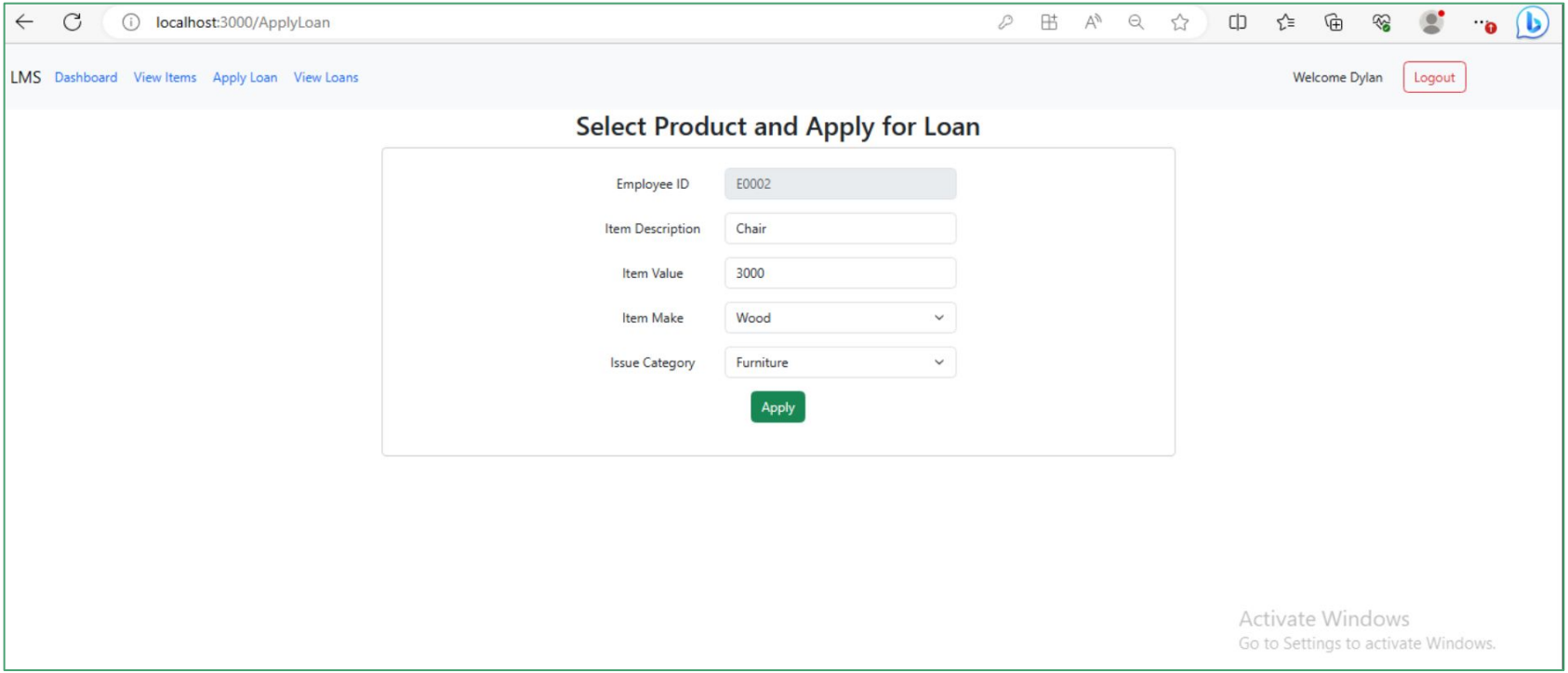
Customer Dashboard



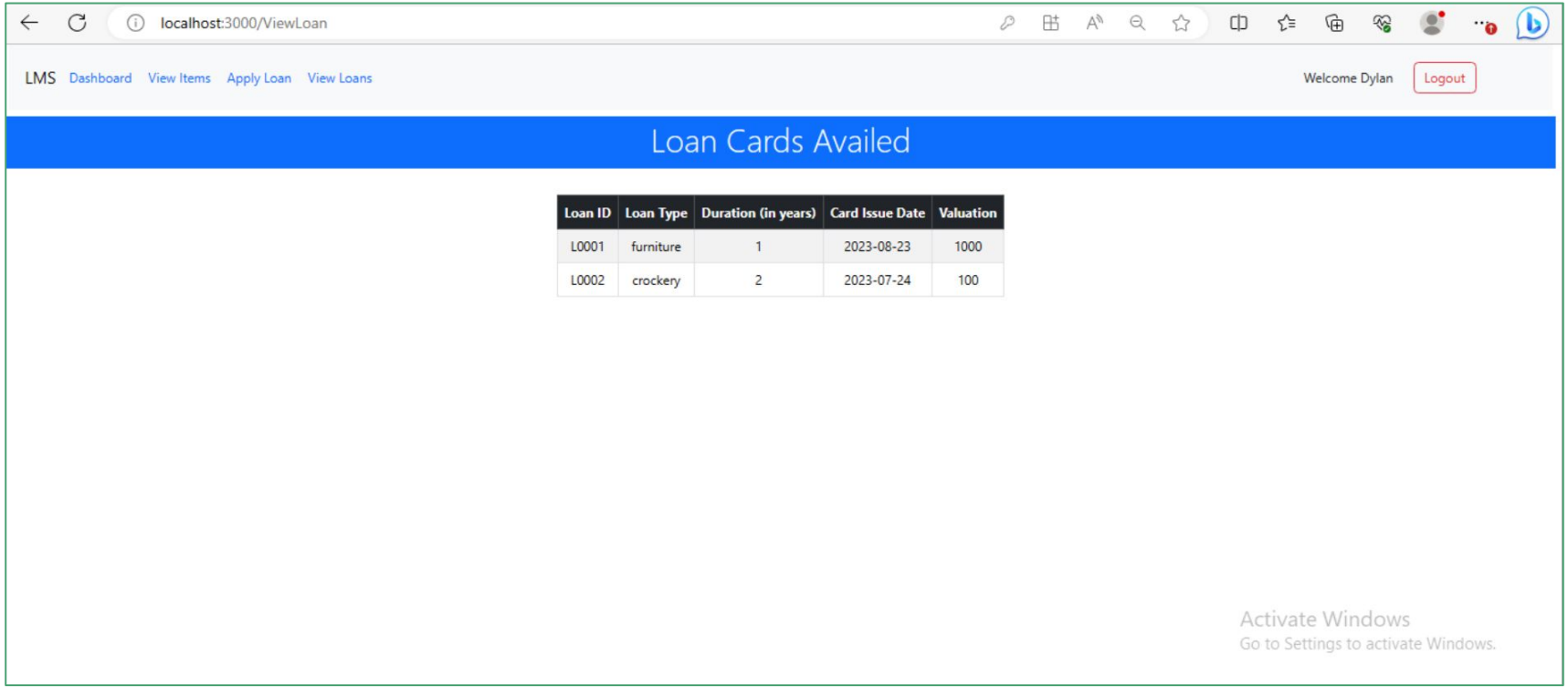
Customer > View Items Purchased



Customer > Apply for Loan



Customer > View Loans



Customer > View Loans (No Loan Cards Availed)