**UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF HIGH QUALITY TRAINING**



**Instructor: Mr. Nguyễn Đức Khoan**

**WEB PROGRAMMING**

**PROJECT REPORT**

**BOOK RENTAL STORE**

**Ngũ Quốc Phong – 17110060**

**Trần Minh Quang – 17110067**

**1st SEMESTER, SCHOOL YEAR: 2019 – 2020**

**HO CHI MINH CITY – 15/12/2019**

# INDEX

Contents

[INDEX 1](#_Toc27790740)

[I. Description 2](#_Toc27790741)

[a. Project Overview: 2](#_Toc27790742)

[b. Data, input 2](#_Toc27790743)

[II. Task allocation 3](#_Toc27790744)

[III. The appication: 3](#_Toc27790745)

[1. Overview 3](#_Toc27790746)

[2. Data Manipulation 4](#_Toc27790747)

[3. Security 5](#_Toc27790748)

[4. Interface 5](#_Toc27790749)

[5. Database design 7](#_Toc27790750)

[IV. Conclusion 9](#_Toc27790751)

## Description

### Project Overview:

This project is designed as a Book Rental System, this system allows users rent books from the database and also allows adminstrator to modify the system as they see fits. The following is used in the application

Technology used:

* ASP.NET MVC as the main Web Application Framework.
* ADO.NET Entity Framework as a support technology.

### Data, input

The program will receive data from the designed Microsoft SQL Server database and the program will receive input from user to interact with the database.

## Task allocation

Tran Minh Quang 17110067:

* Basic Controllers Implementation.
* Database design/Coding.
* CSS Coding.

Ngu Quoc Phong 17110060:

* Project Guidance.
* Controllers Support (On View Model, Session).
* Web/UI Design.

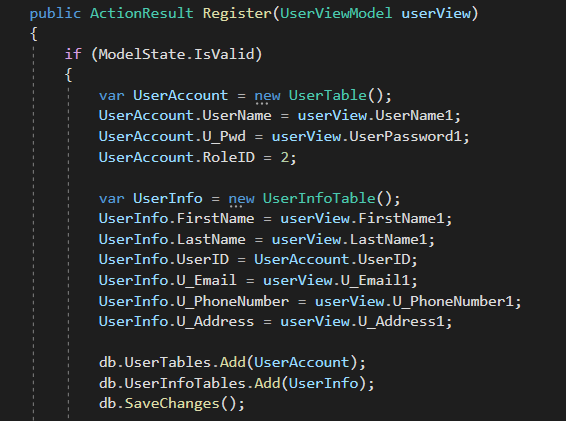
## The appication:

### Overview

Inside this app there are multiple controllers, in each one of them have both actions that are available to the Admin Only, User Only or Both. Each action was configured using methods of verification on a very basic level.

### Data Manipulation

While Entity Framework provided ways to access various information of the database, we sometimes would hit a road block on certain information received and information needed to be post to the server. To overcome this, the concept of View Model was ultilise as a mean to Post info as also a way to Get info to provide users informations that needed to engage in the application.



1: View Model In Action

### Security

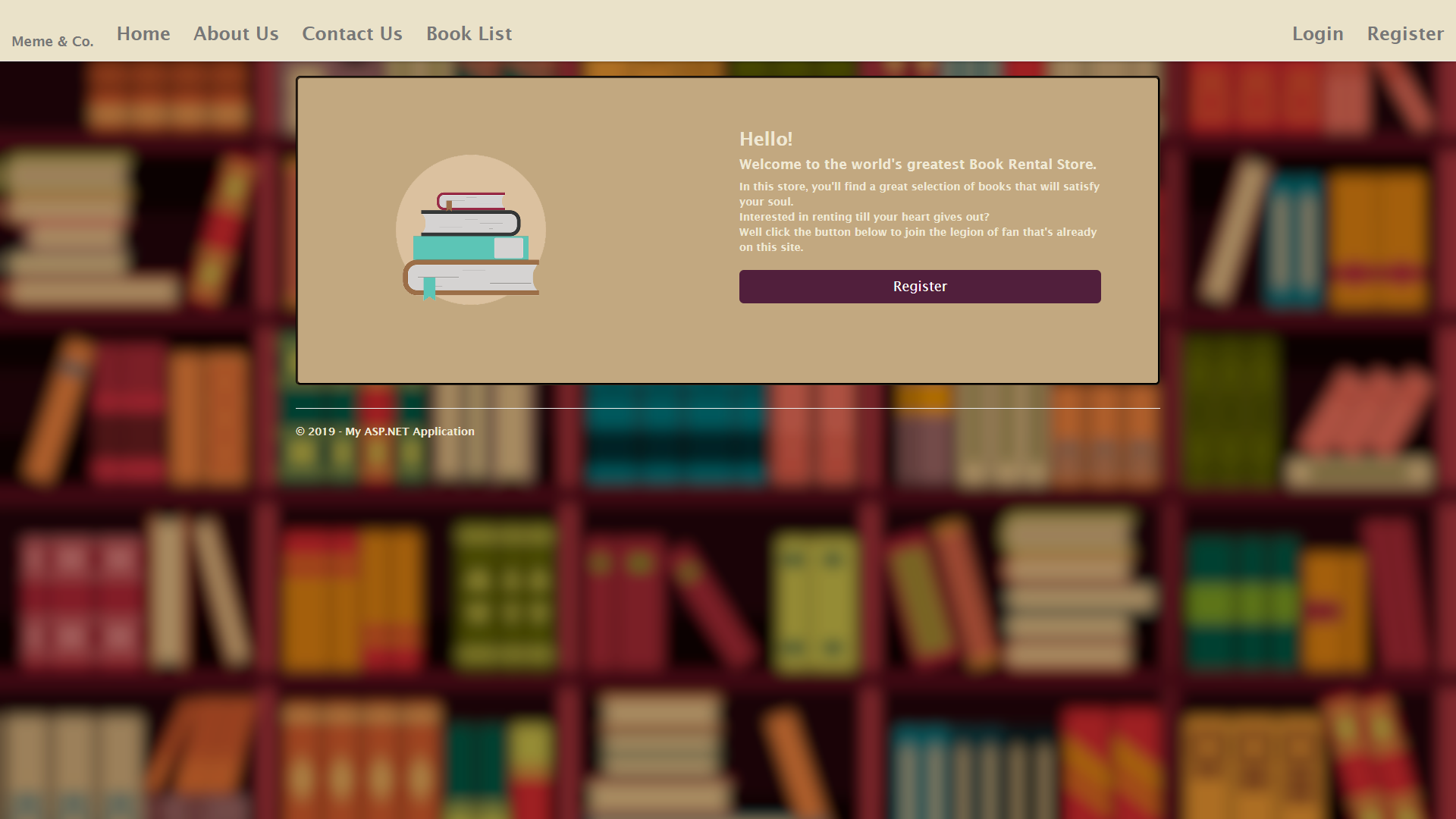
These sessions were used to determined whether user is logged in or is an admin, this of course exposes the website to various means to attack. ASP.NET Identity could’ve been used to ensure security to be at a competent level but sadly due to time constraint we couldn’t spend time to learn to how to ultilise it.



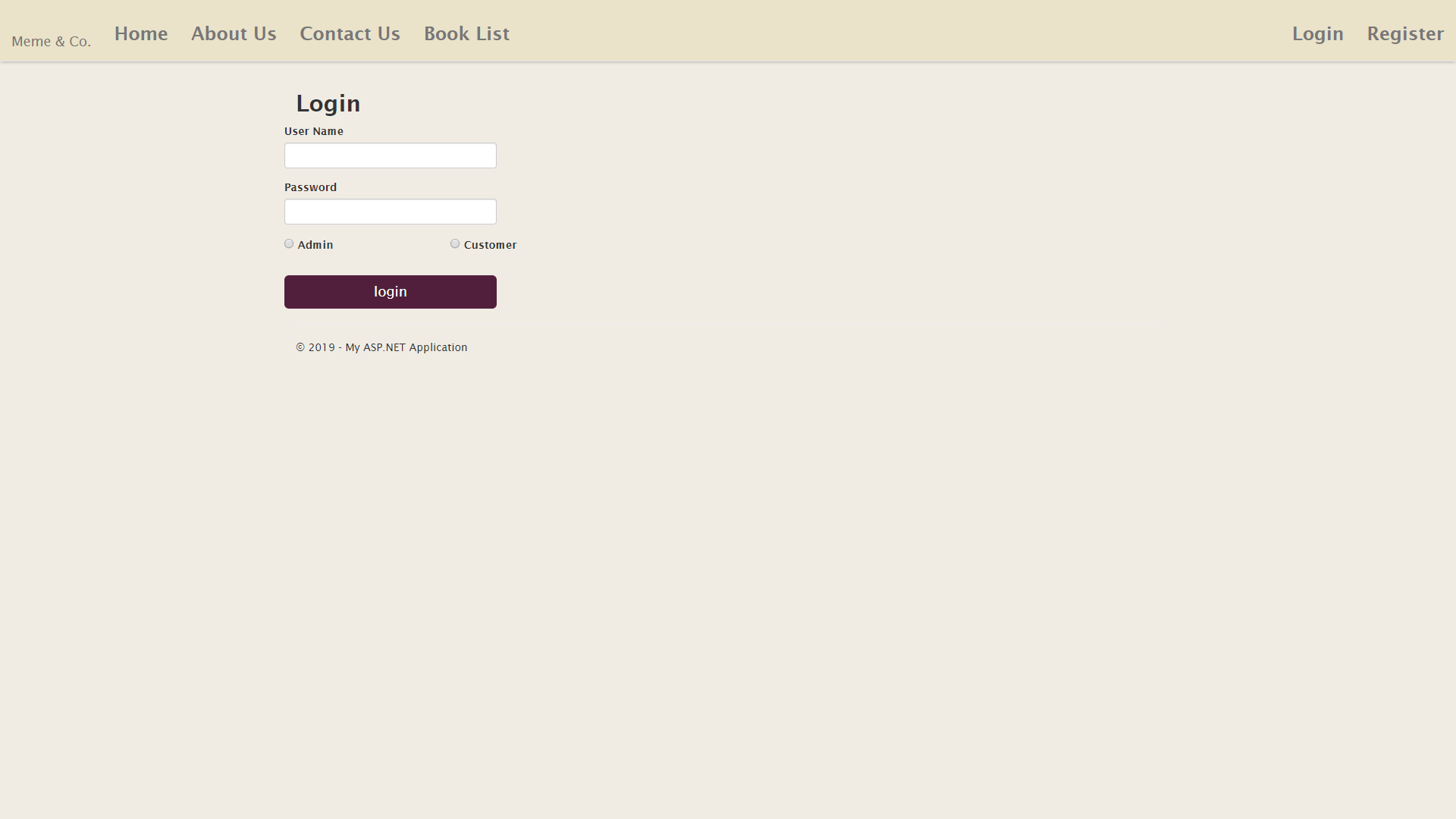
2: Session as a workaround

### Interface

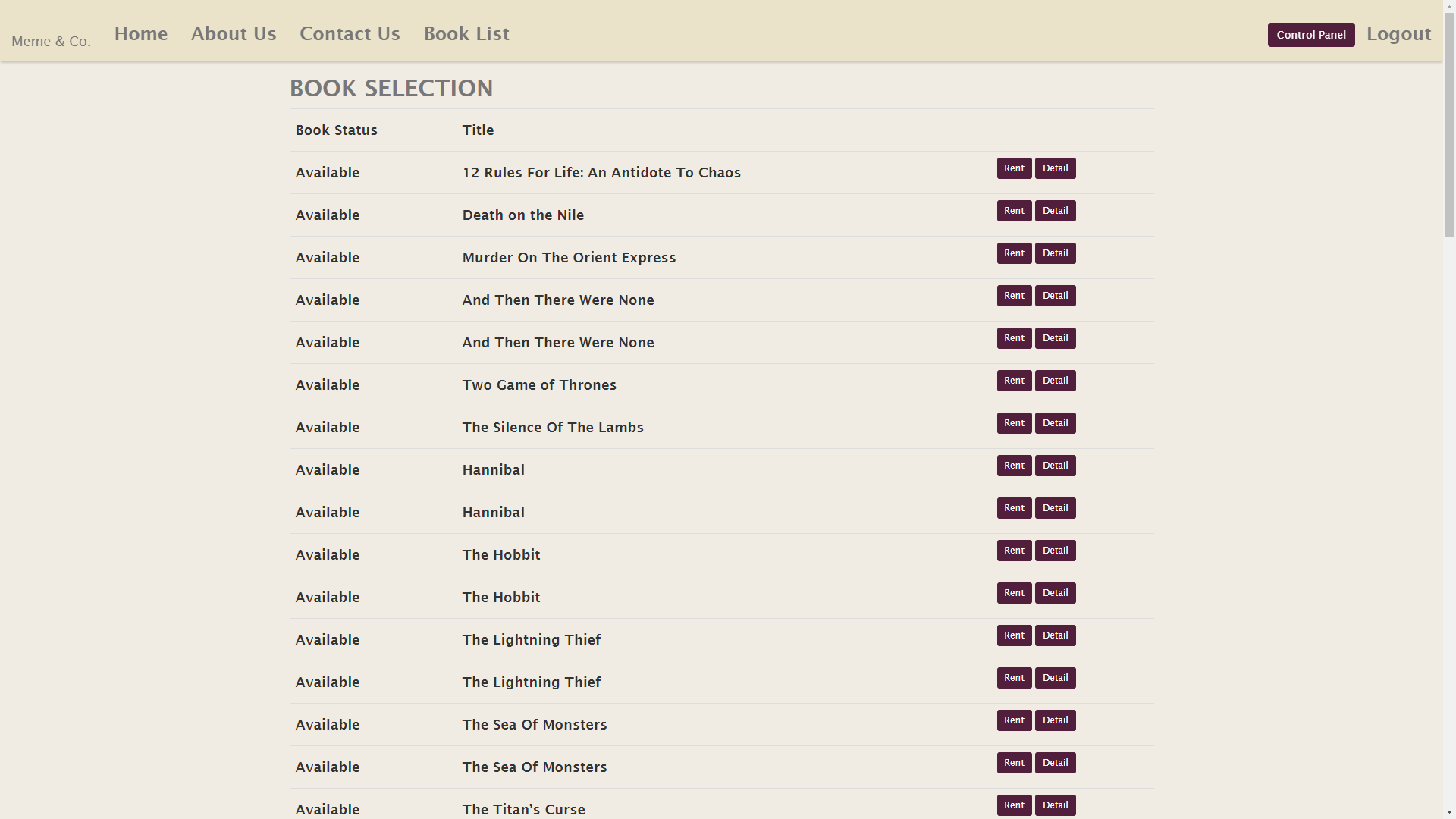
80% of the cshtml were created through scaffolding, we of course would try to make it as professional looking as possible, color scheme and certain aspect were inspired by other popular websites, we of course couldn’t achieve close to what they had done. Opting to basic CSS editing, and some image editing we managed to achieve these result.



3: Front Page of the Website



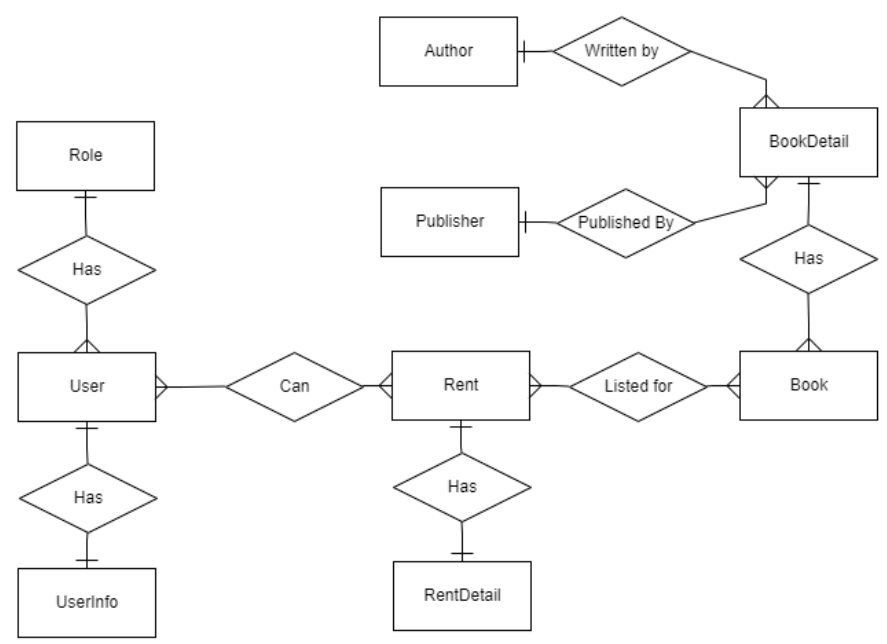
4: Login Page



5: Book Renting Section

### Database design

#### Entity – Relationship Diagram



#### Table description

|  |  |  |
| --- | --- | --- |
|  | Table name | Explaination |
| 1 | RoleTable | Hold Role information |
| 2 | UserTable | Hold Login information |
| 3 | UserInfoTable | Hold User’s detailed information |
| 4 | AuthorTable | Hold Authors’ information |
| 5 | PublisherTable | Hold Publishers’ information |
| 6 | BookDetailTable | Hold Book titles’ information |
| 7 | BookTable | Hold Book list in stock |
| 8 | RentTable | Hold Webpage’s rent order |
| 9 | RentDetailTable | Hold Rents’ detailed information |

## Conclusion

The application behaves as it should, allowing customer to perform simple login and registration. Divided for two type of user, customer and admin, both have their own set of control panel with customer of course limited to their own information only while the administrator have access to everything except some limitation that needed to be solved with the Database manager (In this case SSMS).

While the design wasn’t as beautiful or intuitive as we want but ultimately we managed to put out something that is at least, servicable.

Overall despite being rushed within 7 days, the project came out pretty decently made, despite having a lot of dirty codes, dangerous workarounds and hard coded parts, expect bugs.

References:

* MVC News Project by Mr. Nguyen Duc Khoan.
* Image sources belong to their respectful owner we do not own any of the used images.
* Microsoft of documentation of various topics (MVC, Entity Framework).
* W3School HTML/CSS Tutorials.