

Table of Contents

User Manual (Instruction of running program)	1
Implementation of functions	10
Software Architecture	12
Description of methods of Class.....	13
Work Experience in ASP .NET CORE MVC.....	16
Bibliography	17

List of Figures

Figure 1 Sign in/Signup screen	1
Figure 2 Dashboard of Ropey DVD System	2
Figure 3 Adding actor record	3
Figure 4 Searching DVDs of a chosen actor by last name	3
Figure 5 Viewing available copies of DVDs chosen by actor's last name	4
Figure 6 Viewing last 31 days loan history of a selected member	4
Figure 7 Viewing detailed information of DVDs	5
Figure 8 Viewing last loan of a DVD Copy	5
Figure 9 Viewing all the loan history	6
Figure 10 Viewing member's information	6
Figure 11 Adding DVD record	7
Figure 12 Interface to manage user roles	8
Figure 13 Assigning role to any user	8
Figure 14 Interface to change password of a user	9
Figure 15 MVC Architecture	12

List of Tables

Table 1 Concise description of functionalities	11
Table 2 Method and its description	15

User Manual (Instruction of running program)

To run the program with dummy data, a text file for seeding is provided. Using that file, database can be seeded and program can be run from Visual Studio or with command 'dotnet run'.

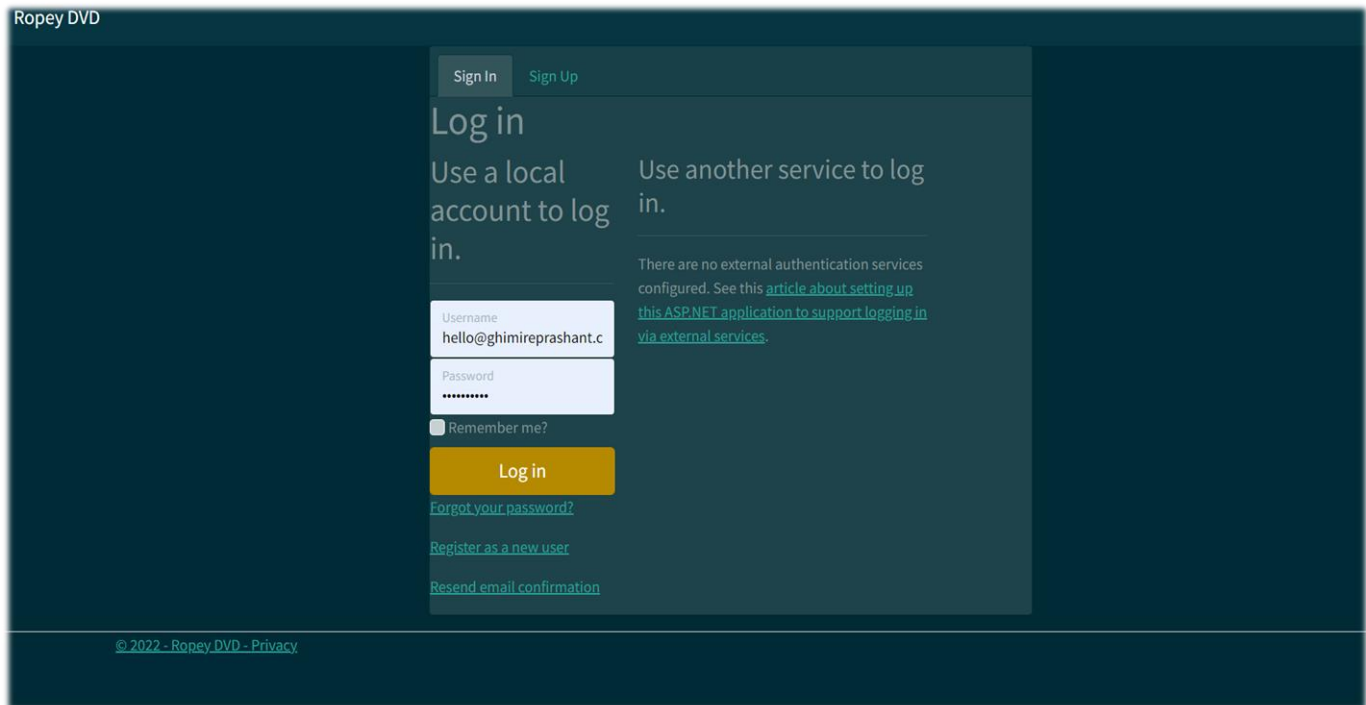


Figure 1 Sign in/Signup screen

After signing in or signing up to the program, user will be navigated to the dashboard as shown in next page.



Figure 2 Dashboard of Ropey DVD System

From the dashboard, user can easily perform their work reading the instructions shown in the user interface.

Record creation for actor, producer and other adding to the table records can be done through top navigation bar. Card interface in the home page can be used to perform asked functionalities of the provided question.

The screenshot shows the 'Create Actor' form in the Ropey DVD application. The form has a dark teal background with white text. At the top, there is a navigation bar with links: 'Ropey DVD', 'Actors', 'Producers', 'DVDs', 'MemberShip Categories', 'RoleManager', and 'Old Copies'. Below the navigation bar, the title 'Create Actor' is displayed. The form contains two input fields: 'ActorSurname' with the value 'Hamal' and 'ActorFirstName' with the value 'Rajesh'. Below these fields is a yellow 'Create' button and a blue link 'Back to List'. At the bottom, there is a copyright notice: '© 2022 - Ropey DVD - Privacy'.

Ropey DVD Actors Producers DVDs MemberShip Categories RoleManager Old Copies

Create Actor

ActorSurname
Hamal

ActorFirstName
Rajesh

Create

[Back to List](#)

© 2022 - Ropey DVD - Privacy

Figure 3 Adding actor record

The screenshot shows the 'ActorDVD' page in the Ropey DVD application. The page has a dark teal background with white text. At the top, there is a navigation bar with links: 'Ropey DVD', 'Actors', 'Producers', 'DVDs', 'MemberShip Categories', 'RoleManager', and 'Old Copies'. Below the navigation bar, there is a search bar with the text 'Actor's last name: hamal' and a 'Search' button. Below the search bar is a table with two columns: 'Actor' and 'Movie'. The table contains four rows of data. At the bottom, there is a copyright notice: '© 2022 - Ropey DVD - Privacy'.

Ropey DVD Actors Producers DVDs MemberShip Categories RoleManager Old Copies

Actor's last name: hamal Search

Actor	Movie
Rajesh Hamal	Kabadi kabadi
Rajesh Hamal	Inception
Rajesh Hamal	Forest Gump
Rajesh Hamal	Into the wild

© 2022 - Ropey DVD - Privacy

Figure 4 Searching DVDs of a chosen actor by last name

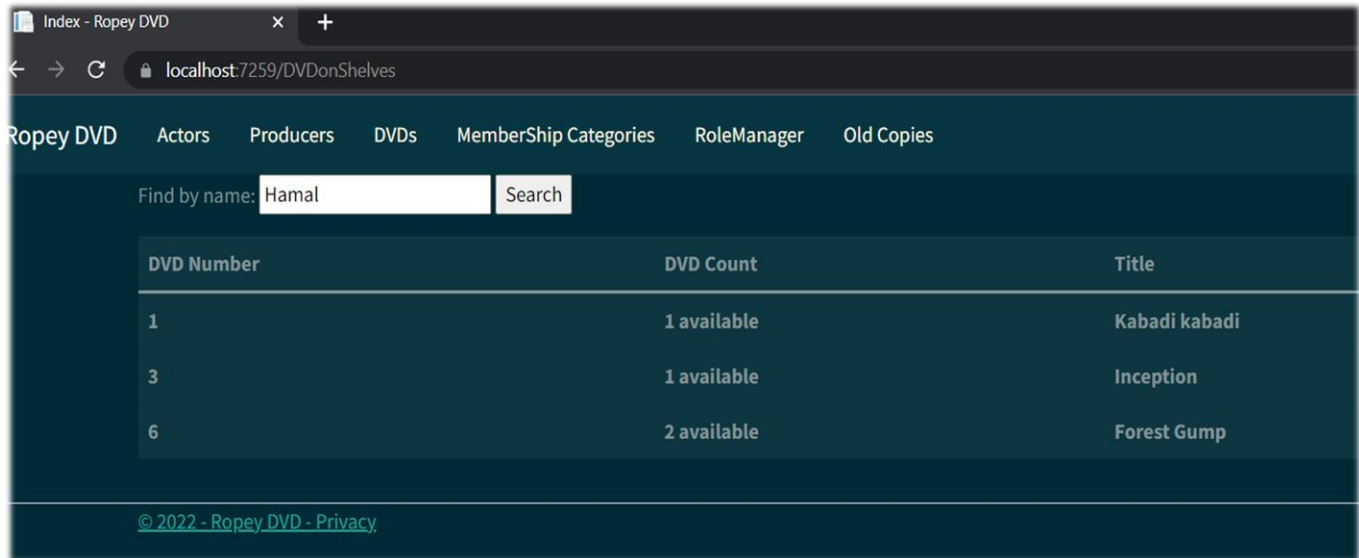


Figure 5 Viewing available copies of DVDs chosen by actor's last name

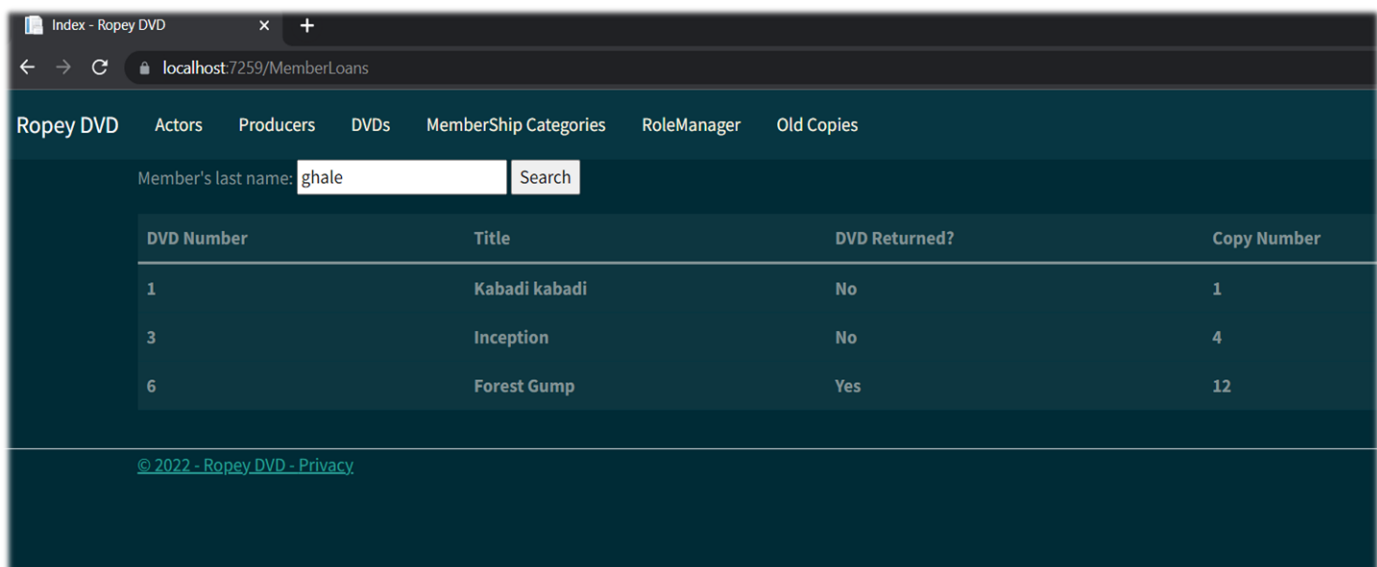


Figure 6 Viewing last 31 days loan history of a selected member

DVD	Release	Standard Charge	Penalty	Studio	Producer	Cast Members
Inception	6/23/2011 12:00:00 AM	150	200	Disney	Nischal bashnet	Rajesh Hamal, Bhuban K.C.
Into the wild	6/23/2012 12:00:00 AM	90	100	Pixar	Kanye West	Rajesh Hamal, Anmol K.C., Nikhil Upreti
Spiderman	7/3/2013 12:00:00 AM	100	200	Hilights Nepal	Quintin Tarantino	Leo Di caprip, Nikhil Upreti
Raman Ragav	5/23/2015 12:00:00 AM	500	600	Hilights Nepal	Martin Scorsese	Daya Hang Rai, Bhuban K.C.
Forest Gump	6/23/2018 12:00:00 AM	50	100	Hilights Nepal	Martin Scorsese	Rajesh Hamal
Kabadi kabadi	6/23/2018 12:00:00 AM	50	100	Hilights Nepal	Martin Scorsese	Rajesh Hamal, Saugat Malla

© 2022 - Ropey DVD - Privacy

Figure 7 Viewing detailed information of DVDs

Enter copy number: 4 Search

Last Loan

DVD:	Inception
Borrower/Member:	Samrat Ghale
Returned?	No
LoanNumber	4
DateOut	4/15/2022 12:00:00 AM
DateDue	5/15/2022 12:00:00 AM
DateReturned	
LoanTypeNumber	2
CopyNumber	4
MemberNumber	1

Return DVD

© 2022 - Ropey DVD - Privacy

Figure 8 Viewing last loan of a DVD Copy

Ropey DVD Actors Producers DVDs MemberShip Categories RoleManager Old Copies Profile Logout

[Create New](#)

DateOut	DateDue	DateReturned	DVDCopy	LoanType	Member	
4/12/2022 12:00:00 AM	5/12/2022 12:00:00 AM		1	1	Samrat	Edit Details Delete
4/13/2022 12:00:00 AM	5/13/2022 12:00:00 AM	5/8/2022 12:00:00 AM	2	1	Bishant	Edit Details Delete
4/14/2022 12:00:00 AM	5/14/2022 12:00:00 AM		3	2	Ishan	Edit Details Delete
4/15/2022 12:00:00 AM	5/15/2022 12:00:00 AM		4	2	Samrat	Edit Details Delete
4/16/2022 12:00:00 AM	5/16/2022 12:00:00 AM		5	1	Ishan	Edit Details Delete
4/17/2022 12:00:00 AM	5/17/2022 12:00:00 AM		6	3	Bishant	Edit Details Delete
4/18/2022 12:00:00 AM	5/18/2022 12:00:00 AM	4/28/2022 12:00:00 AM	7	1	Bishant	Edit Details Delete
4/19/2022 12:00:00 AM	5/19/2022 12:00:00 AM	5/11/2022 12:00:00 AM	8	3	Sugam	Edit Details Delete
4/10/2022 12:00:00 AM	5/10/2022 12:00:00 AM	4/10/2022 12:00:00 AM	9	2	Sugam	Edit Details Delete
4/11/2022 12:00:00 AM	5/11/2022 12:00:00 AM	4/11/2022 12:00:00 AM	10	1	Arun	Edit Details Delete
5/8/2022 12:00:00 AM	5/9/2022 12:00:00 AM	5/9/2022 12:00:00 AM	12	2	Samrat	Edit Details Delete
5/8/2022 12:00:00 AM	5/15/2022 12:00:00 AM		14	3	Ishan	Edit Details Delete
5/8/2022 12:00:00 AM	5/9/2022 12:00:00 AM	5/9/2022 12:00:00 AM	17	2	Ishan	Edit Details Delete
5/8/2022 12:00:00 AM	6/7/2022 12:00:00 AM		17	1	Ishan	Edit Details Delete
5/8/2022 12:00:00 AM	5/15/2022 12:00:00 AM		16	3	Ishan	Edit Details Delete

Figure 9 Viewing all the loan history

Assistants, Managers can also edit any particular loan transactions from this interface.

Ropey DVD Actors Producers DVDs MemberShip Categories RoleManager Old Copies Profile Logout

All Members

MemberLastName	MemberFirstName	MemberAddress	MemberDateOfBirth	MembershipCategoryDescription	LoanCount
Bohora	Arun	Kathmandu	2/3/2001 12:00:00 AM	Normal	0
Gautam	Bishant	Kathmandu	10/1/2009 12:00:00 AM	Medium	1
Limbu	Ishan	Kathmandu	3/1/2002 12:00:00 AM	Normal	5
Giri	Rijan	Kathmandu	1/1/2001 12:00:00 AM	Medium	0
Ghale	Samrat	Kathmandu	12/1/1999 12:00:00 AM	Premium	2
Adhikari	Sugam	Kathmandu	7/1/2010 12:00:00 AM	Premium	0

[© 2022 - Ropey DVD - Privacy](#)

Figure 10 Viewing member's information

03 / 05 / 2022 , 07 : 47

StandardCharge

100

PenaltyCharge

300

CategoryDescription

Adventure

Toggle Input

StudioName

Pixar

Toggle Input

ProducerName

Kanye West

Rajesh

Add

ActorFirstName

Create

[Back to List](#)

© 2022 - Ropey DVD - Privacy

Figure 11 Adding DVD record

Users with assistant, manager role can add new DVD record either by choosing data from drop down or using text field if new records for studio, producers are required.

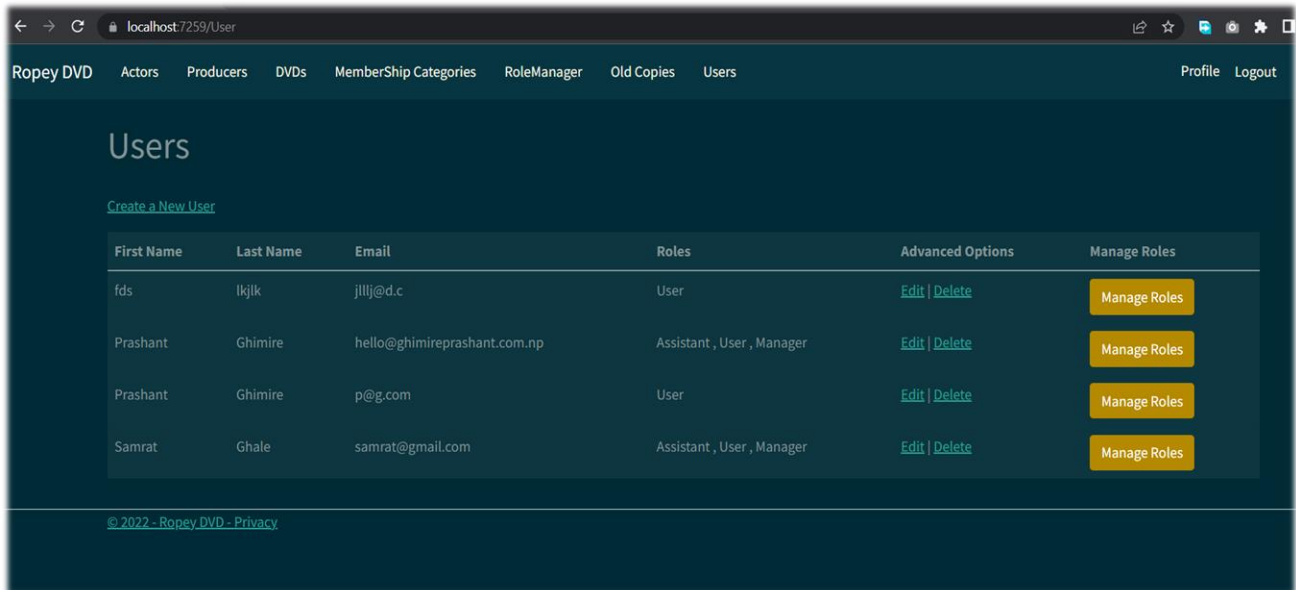


Figure 12 Interface to manage user roles

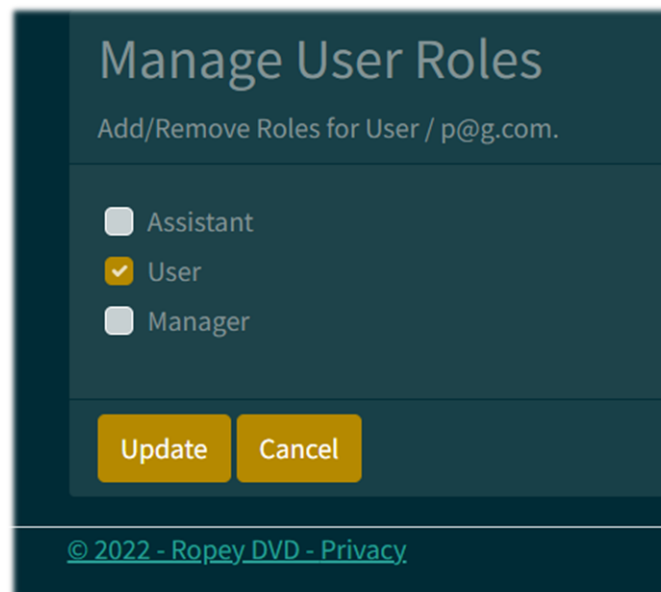


Figure 13 Assigning role to any user

This is the interface of manager from where roles can be assigned to other employees of the shop.

The screenshot displays a user account management interface. On the left, a sidebar lists navigation options: Profile, Email, Password (highlighted), Two-factor authentication, and Personal data. The main content area is titled 'Change password' and contains three input fields: 'Current password', 'New password', and 'Confirm new password', each with a masked password (dots). Below these fields is a prominent yellow 'Update password' button. At the bottom left of the interface, there is a copyright notice: '© 2022 - Ropey DVD - Privacy'.

Figure 14 Interface to change password of a user

The above shown user interfaces is the guide of using implemented features of the application being the user manual documentation.

Implementation of functions

Function	Description
1	From the Cast Member's bridge entity, those DVD number is used with user's chosen actor number. After that, DVD Title information is retrieved with those DVD numbers.
2	Five tables which are DVDTitle, CastMember, DVDCopy, Actor, DVDOnShelves are used to make this function work. To count available copies, if data returned has null value or not in any copies. If null value is present, count of available copies is increased to count total.
3	To return a member's loan within past 31 days, only the loan history from present date to last 31 days is filtered and obtained information was passed to interface.
4	To show all cast members, actors of each DVD are combined and passed in list. Later to show in view, other DVD information like producer, studio is passed along with list of actors.
5	To show the last loan of selected member, first loan information is picked from the descending order list of the loans taken by the member.
6	To add a new loan record, drop down list is provided to user for selecting member and DVD copy. The loan information is then serialized and store in loan table of the database.
7	Before setting DVD returned date to a loan, loan period time is calculated and information is shown on view as per loan period.
8	Member's information is shown along with their total loan count. To count their loans, Loan table is used and repetition of a member in the rows is counted.

9	User can create a new record of DVD with available producers, studio information along with the release date. Apart from drop down records, user can add new records for producers, studio if required.
10	Using this function, assistant and manager can remove the old DVD copies.
11	This function is used to produce those DVDs on loan shown with dates.
12	This function is used to show inactive users who have not borrowed any loans in last 31 days.
13	Using this function, assistant and managers can view those DVD which are not borrowed in last 31 days.
14	Users of the system are allowed to change their account password with this functionality.

Table 1 Concise description of functionalities

Software Architecture

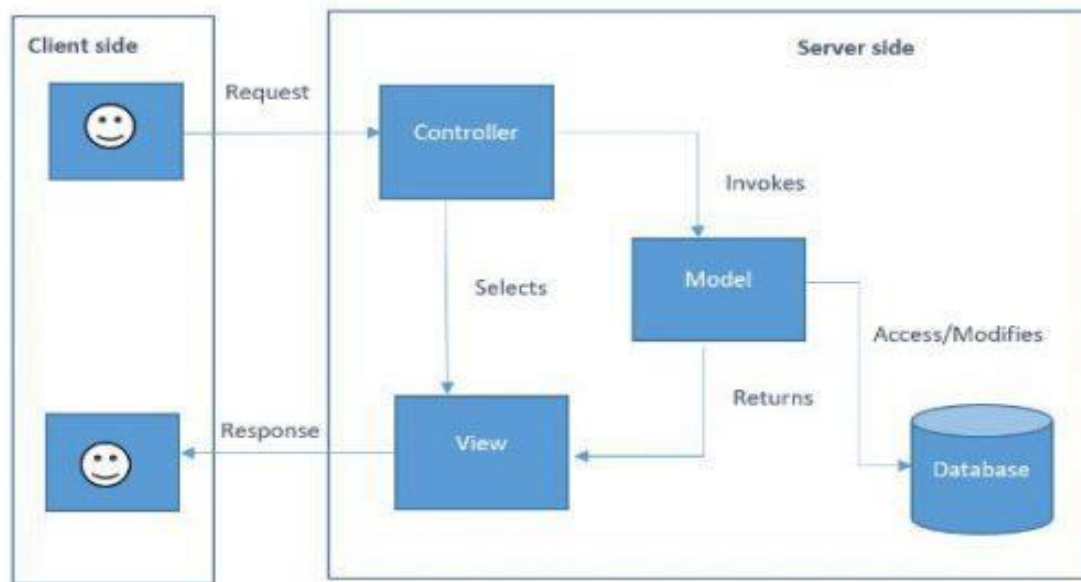


Figure 15 MVC Architecture

Working mechanism of MVC architecture in short:

The model view controller architecture pattern serves the front-end user requests of the system. Controller classes of each functions handles the user's request and using the Model class, database is connected and essential data is flown between controller and database using Model class. Then Controller class passes information to View which is then obtained in user friendly manner in the interface. This way, a MVC architecture works in short (Deacon, 2016).

Description of methods of Class

To fulfil the requirements of the coursework, several classes were constructed while some of the classes were auto-generated as built-in ASP .NET functionalities.

Auto-generated classes:

- ActorsController
- DVDCategoriesController
- DVDCopiesController
- DVDTitlesController
- HomeController
- LoansController
- LoanTypesController
- MembersController
- MembershipCategoriesController
- ProducersController
- StudiosController

Self-created classes to implement functionalities from requirements:

- ActorDVDController
- DVDDetailsController
- DVDLoanController
- DVDOnShelvesController
- DVDReturnController
- MemberDetailsController
- MemberLoansController
- OldCopiesController
- RolManagerController
- UserRolesController

Common methods in controllers:

Method	Description
<code>public async Task<IActionResult> Index()</code>	It returns records of desired data to the view or user interface. The table list of producers, actors, studio, DVD details are rendered using this method.
<code>public async Task<IActionResult> Details(int? id)</code>	It returns a detailed view of a record of a particular row when 'Details' button is clicked for a particular row of data.
<code>public async Task<IActionResult> Create([Bind("Properties of class as parameter")] DataModel modelInstance)</code>	This same format is used to create methods to insert data in the table. For example: if we want to add new actor record, the function will be <code>public async Task<IActionResult> Create([Bind("ActorNumber, ActorSurname, ActorFirstName")] Actor actor)</code> In this same format, other classes has their own create method but the format is same for create operation.
<code>public async Task<IActionResult> Edit(int? id)</code>	This method is used in the controller classes where there is need to edit the information of a row/tuple. For example: editing actor, producer, studio and similar records, this function is used in our system.
<code>public async Task<IActionResult> Delete(int? id)</code>	This method is used to return the view to delete user's chosen row from the table.

<code>public async Task<IActionResult> DeleteConfirmed(int id)</code>	This method confirms the deletion of a row from the table and returns new updated view in the interface.
<code>private async Task<List<string>> GetUserRoles(ApplicationUser user)</code>	This method is used to retrieve user roles from the database.
<code>public UserRolesController(UserManager<ApplicationUser> userManager, RoleManager<IdentityRole> roleManager)</code>	This method is used to make changes in the roles of users where manager can assign or remove roles to the users of the system.

Table 2 Method and its description

Work Experience in ASP .NET CORE MVC

This is my first time working in MVC architecture and found that auto generation of some parts of CRUD operations were really time saving from my perspective. I was only familiar with micro-framework like Flask and after using ASP .NET Core MVC, I found this framework very effective to carry out CRUD functionality web applications. Language-Integrated Query (LINQ) which we used was similar to the SQLAlchemy that I previously used in Flask.

The NuGet packages like SQL Server, User Identity were useful and I found the shared layout feature very handy for my part of task because, with this, there was no need to make changes in every static html content of other pages. While working with CSS, I felt task redundancy. To minimize it, I used a solarized bootstrap theme for overall project.

Bibliography

Deacon, J., 2016. *Rare Parts*. [Online]

Available at: <https://www.rareparts.com/pdf/MVC.pdf>

[Accessed 27 April 2022].