**Library Management**

**By**

**Jitendra Adtani**

Login Page:

The first window when application runs is the login page. User will not be able to navigate to other pages until he logs in with a registered email. If email doesn’t exist in the database, it will show below error message. A user can click on register to register himself, and the email will be added in the database. I might have forgot to put the [Email] validation attribute back in the model which will only allow type emails in textbox.

Graphical user interface, website

Description automatically generated

On successful login, user will be redirected to the Add Books window. Empty book name, author and quantity will not be allowed to insert. The validations will show the error message as required. All Validations have been applied in the models with [Required(ErrorMessage=”Requried!”] attribute.

Graphical user interface, website

Description automatically generated

All books tab shows below window with all the available and not available books. If quantity is 0, the status will show Not Available.

Graphical user interface, website

Description automatically generated

Clicking on borrow shows success pop-up message, and books get added in the borrow database which can be seen in My Books window. Same goes for return functionality, on successful return, a pop-up message will show successfully borrowed.

Graphical user interface, website

Description automatically generated

My Books window shows all the created and borrowed books by that user.

A screenshot of a computer

Description automatically generated with medium confidence

User can delete or edit the added books and can return the borrowed books at the bottom. Both the tables show only books belonging that user.

Sort actions can be performed by selecting the radio button and clicking on search button. Here, you can see, I searched for a Book Name with a name ‘Moby’, and it shows the results below.

Graphical user interface, text, application, email, website

Description automatically generated

We can also sort by ascending and descending orders by clicking the row headers. A book can also be return from the All-books window. Once a book has been borrowed, the borrow button will be disabled for that user until he/she returns the book. Once returned, the borrow button will be activated again.

Clicking on logout button in navigation bar will end the session for the current user, and the user will be redirected to the login page.

A screenshot of a computer

Description automatically generated with medium confidence

Setup:

To setup the application, open the visual studio solution file and open SQL server management studio. Run the SQL script that has come with solution. It will create all the tables and users. Once finished, attach the database in visual studio with the help of connection string. Users can also be created from the front-end GUI by clicking on Register button.

Graphical user interface, text, application, email

Description automatically generated

Once connected with database, create a ‘context’ folder in solution explorer (if doesn’t exist) and add an entity model item by selecting all the tables.

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

The relation model will look something like this,

A screenshot of a computer

Description automatically generated with medium confidence

There will be a few errors as the name of the entity model has changed, the controllers who had instances of it, need to be pointed to the new model name.

Text

Description automatically generated

Do same for the LoginController.

There is one more place where this will be requried. In BookList.cshtml file,

A screenshot of a computer

Description automatically generated with medium confidence

That’s it. The solution now should run without any errors.