



eBook Reader

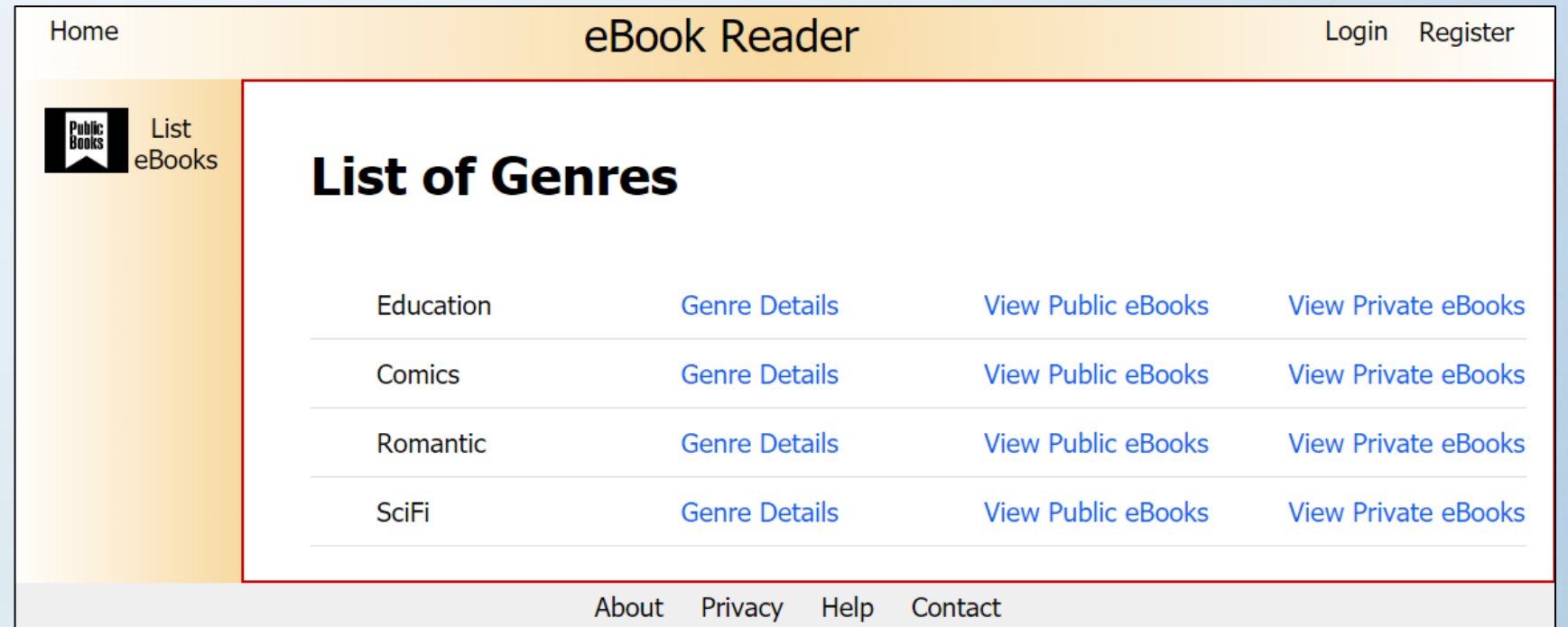
Omkar Buchade

Implementations

- Developed a website with two sections: public section and private section.

Details:

- The public section consists of publicly available eBooks which can be accessed by any user whereas the private section consists of eBooks which need authentication to read them.



Implementations

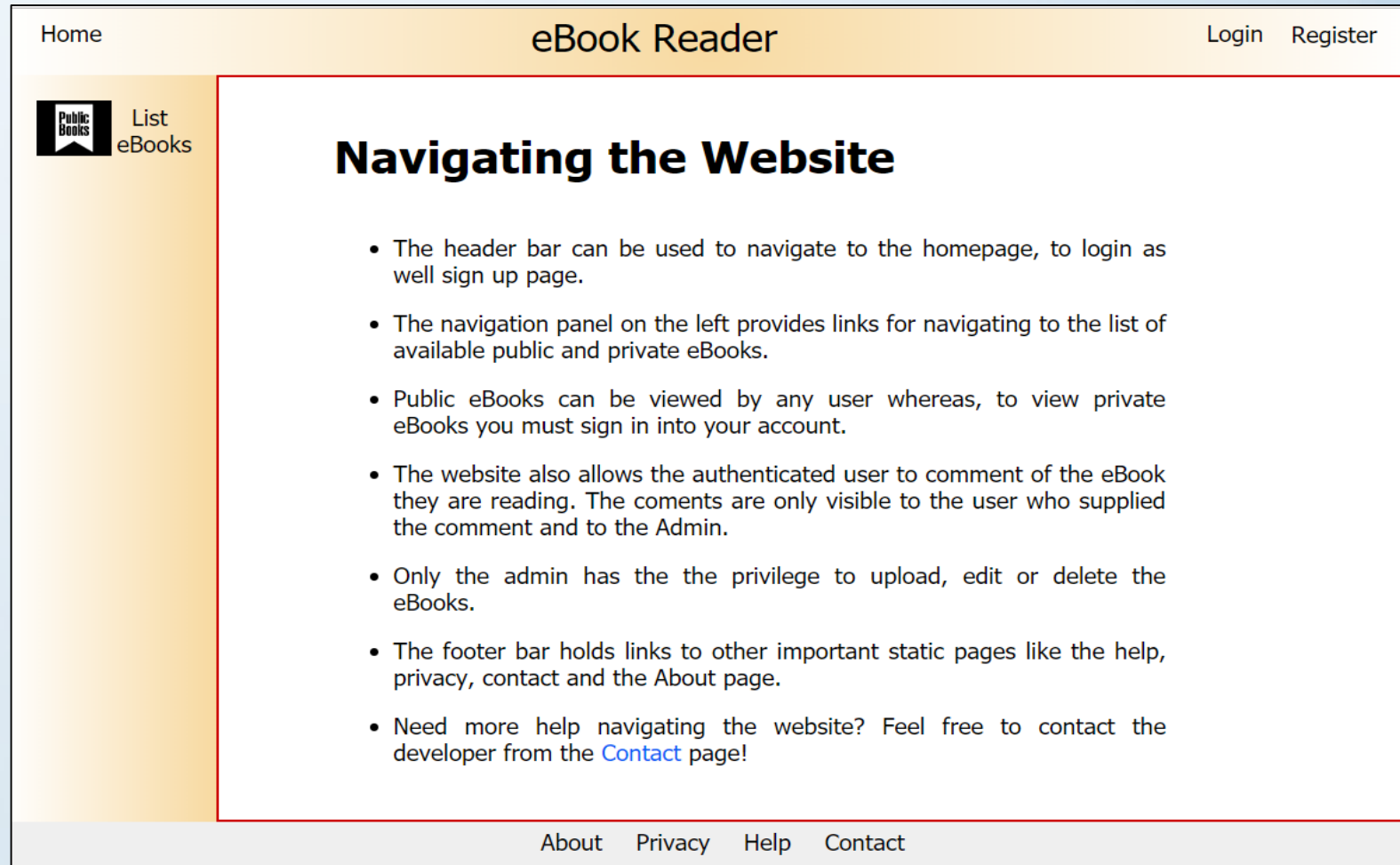
- Used static pages for the public area with HTML5, CSS3 and ES6 for anonymous users.

New features used:

HTML5: Header tag

CSS3: box-shadow,
border-radius,
background-image:
linear-gradient

ES6: Class



Implementations

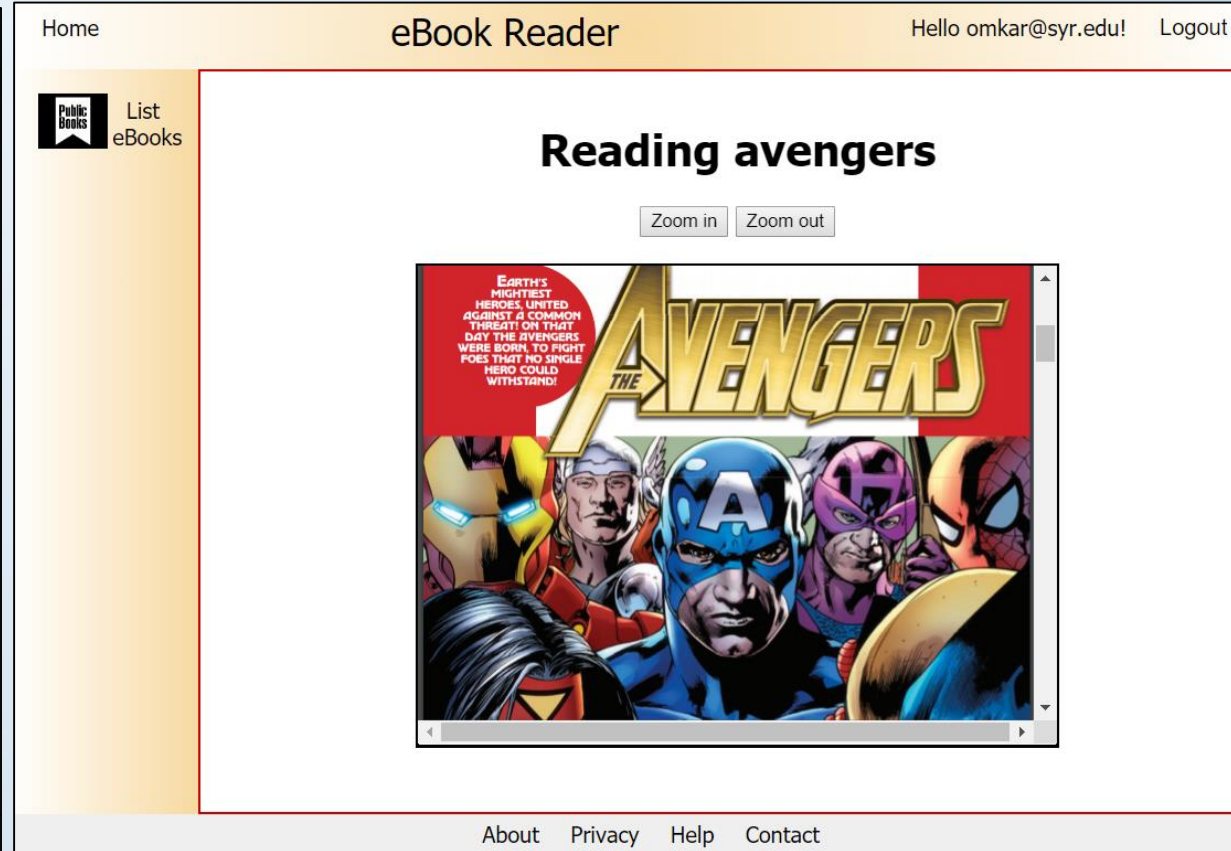
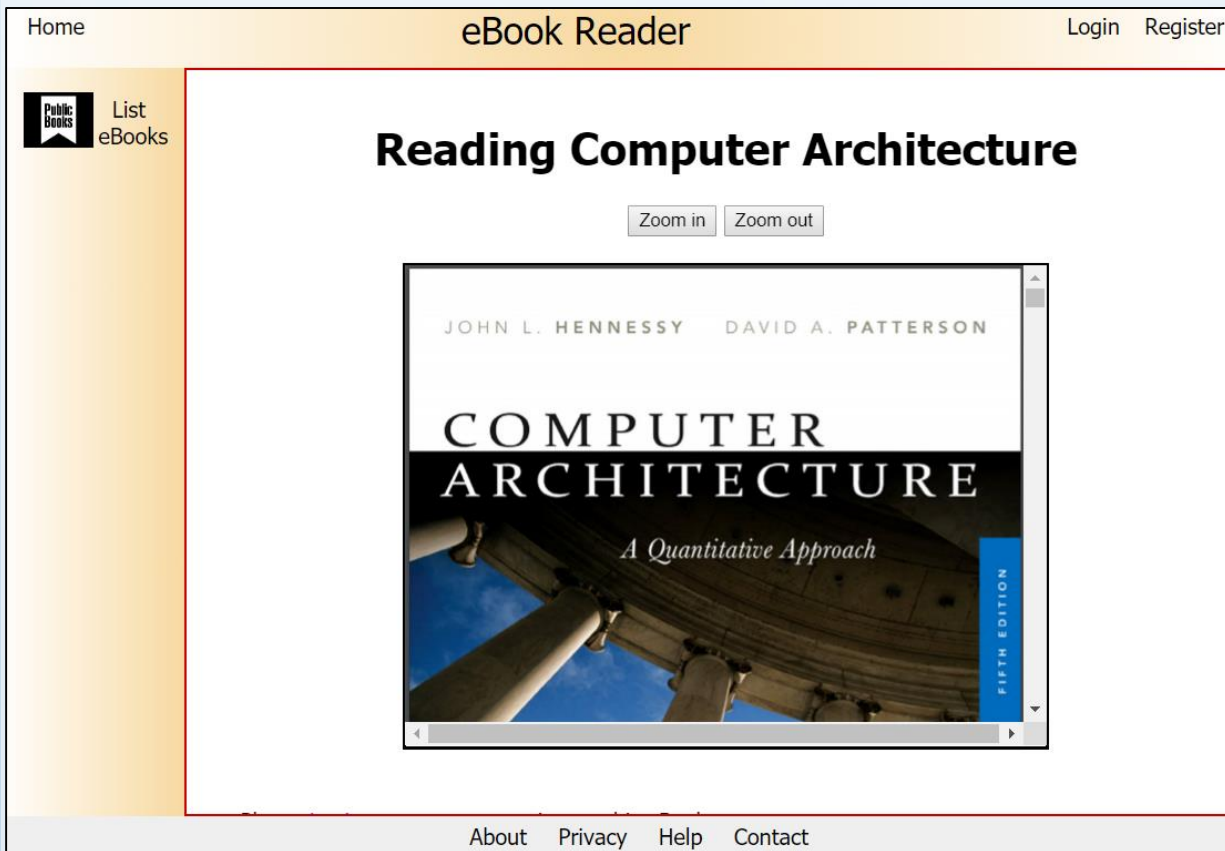
- Used Entity-Framework based Authentication and Authorization.

```
1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // ApplicationDbContext.cs -DbContext class for eBook reader.//
3 //
4 // Omkar Buchade, CSE686 - Internet Programming, Spring 2019 //
5 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
6 /*
7  * This code creates a DbSet property for each (Genre, Book and Book Comment) ent
8  */
9 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
10 using Microsoft.EntityFrameworkCore;
11 using WebApplication.Models;
12 namespace WebApplication.Data
13 {
14     14 references
15     public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
16     {
17         0 references | 0 exceptions
18         public ApplicationDbContext() { }
19         0 references | 0 exceptions
20         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options
21             : base(options)
22         {
23         }
```

```
1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 // IdentityModel.cs - IdentityModel class for eBook reader. //
3 //
4 // Omkar Buchade, CSE686 - Internet Programming, Spring 2019 //
5 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
6 /*
7  * - This package help us to create our own
8  *   identity model by inheriting the IdentityUser
9  *   and simply add the fields which we need.
10  */
11
12 using Microsoft.AspNetCore.Identity;
13 using System.ComponentModel.DataAnnotations;
14 namespace WebApplication.Models
15 {
16     16 references
17     public class ApplicationUser : IdentityUser
18     {
19         //Adding a string type new field while creating a new account.
20         [Required]
21         2 references | 0 exceptions
22         public string Name { get; set; }
23     }
24 }
```


Implementations


- User has the ability to read books in the public section, and with authentication in the private section.



Implementations

- Books have viewing controls that can be used to scale and iterate through the document.

[Home](#) [eBook Reader](#) Hello omkar@syr.edu! [Logout](#)

 [List eBooks](#)

Reading Cpp

[Zoom in](#) [Zoom out](#)

Overloaded functions.

In C++ two different functions can have the same name if their parameter types or number are different. That means that you can give the same name to more than one function if they have either a different number of parameters or different types in their parameters. For example:

```
// overloaded function
#include <iostream>
using namespace std;

int operate (int a, int b)
{
    return (a*b);
}

float operate (float a, float b)
{
    return (a/b);
}

int main ()
{
    int a=3, b=2;
    float m=3.0, n=2.0;
    cout << operate (a,b);
    cout << "\n";
    cout << operate (n,m);
    cout << "\n";
    return 0;
}
```

In this case we have defined two functions with the same name, `operate`, but one of them accepts two parameters of type `int` and the other one accepts them of type `float`. The compiler knows which one to call in each case by examining the types passed as arguments when the function is called. If it is called with two ints as its arguments it calls to the function that has two `int` parameters in its prototype and if it is called with two floats it will call to the one which has two `float` parameters in its prototype.

In the first call to `operate` the two arguments passed are of type `int`, therefore, the function with the first prototype is called; This function returns the result of multiplying both parameters. While the second call passes two arguments of type `float`, so the function with the second prototype is called. This one has a different behavior: it divides one parameter by the other. So the behavior of a call to `operate` depends on the type of the arguments passed because the function has been overloaded.

Notice that a function cannot be overloaded only by its return type. At least one of its parameters must have a different type.

[About](#) [Privacy](#) [Help](#) [Contact](#)


Implementations

- Authenticated user can comment of the document for reviewing. This comment is only visible to the user who supplied the comment and to the administrator.

[Home](#)

eBook Reader

Hello omkar@syr.edu! [Logout](#)

 [List eBooks](#)

Add Comment

Enter comment

This book is a great resource

Add


[Return to reading book](#)

[About](#) [Privacy](#) [Help](#) [Contact](#)

[Home](#)

eBook Reader

Hello omkar@syr.edu! [Logout](#)

 [List eBooks](#)

Book Comments

Username	Date	Comment	
omkar@syr.edu	02 May 2019	This book is a great resource	Delete Comment


[Return to reading book](#)

[About](#) [Privacy](#) [Help](#) [Contact](#)

Implementations

- Server-side validation

[Home](#) [eBook Reader](#) Hello omkar@syr.edu! [Logout](#)

 [List eBooks](#)

Edit eBook

Computer @

Please enter valid Book title

Book Author

The Book Author field is required.

ISBN

The Isbn field is required.

Private ☐

Leave unchecked for public book

Edit

[Back to Genre List](#)

[About](#) [Privacy](#) [Help](#) [Contact](#)

```
public class Book
{
    4 references | 0 exceptions
    public int BookID { get; set; }

    [Required, Display(Name = "Book title")]
    [RegularExpression(@"[a-zA-Z]+[ ]*[a-zA-Z]*", ErrorMessage = "Please enter valid Book title")]
    5 references | 0 exceptions
    public string Title { get; set; }

    [Required, Display(Name = "Book Author")]
    [RegularExpression(@"[a-zA-Z]+[ ]*[a-zA-Z]*", ErrorMessage = "Please enter valid Book Author")]
    5 references | 0 exceptions
    public string Author { get; set; }

    [Required]
    [RegularExpression(@"[0-9]+", ErrorMessage = "Please enter ISBN")]
    5 references | 0 exceptions
    public long Isbn { get; set; }

    7 references | 0 exceptions
}
```


Implementations

- Implemented a Web Service and Web Service client that helps the administrator to upload books from the local file system and store the book metadata into the database.

The screenshot displays the 'eBook Reader' web application. The top navigation bar includes 'Home', 'eBook Reader', 'Login', and 'Register'. A sidebar on the left features a 'Public Books' icon and a 'List eBooks' link. The main content area is titled 'List of public eBooks' and contains a table with the following data:

Book title	Book Author	Isbn
Computer Architecture	Patterson	34654

Below the table is a link: [Back to Genre List](#).

Overlaid on the right is a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. It shows the following interaction:

```
2. Exit
Enter choice: 1

sending request to https://localhost:44305/api/Files
Genre Name: Education GenreID: 3
Genre Name: Comics GenreID: 8
Genre Name: Romantic GenreID: 9
Genre Name: SciFi GenreID: 10
Enter GenreId in which you wish to add book: 3

Enter book title: Computer Architecture

Enter book author: Patterson

Enter book ISBN: 34654


Book Access (Enter 1 for private, or 0 to keep book public: 0

Enter book path (only pdf files supported): C:\\Users\\Omkar\\source\\re
pos\\Project\\books\\ca.pdf
StatusCode: 200, ReasonPhrase: 'OK', Version: 1.1, Content: System.Net.H
ttp.HttpConnection+HttpConnectionResponseContent, Headers:
{
  Transfer-Encoding: chunked
  Server: Microsoft-IIS/10.0
  X-SourceFiles: =?UTF-8?B?QzpcVXNlcnNcT21rYXJcc291cmNlXHJlcG9zXFByb2p1Y
3RcV2ViQXBwbGljYXRpb25cYXBpXEZpbGVz?=
  X-Powered-By: ASP.NET
  Date: Thu, 02 May 2019 06:31:09 GMT
}
Press 1 to continue, 0 to quit:
```

Implementations

- Only the administrator is given the privilege to upload, edit or delete the information related to the document.

[Home](#)eBook Reader[Login](#) [Register](#)

 [List eBooks](#)


List of public eBooks

Book title	Book Author	Isbn	
Comic Novel	Jeff Matt	597	Read Book
Comic History	George Athan	712	Read Book

[Back to Genre List](#)

[About](#) [Privacy](#) [Help](#) [Contact](#)

[Home](#)eBook ReaderHello omkar@syr.edu! [Logout](#)

 [List eBooks](#)

List of public eBooks

Book title	Book Author	Isbn			
Comic Novel	Jeff Matt	597	Read Book	Edit Book	Delete Book
Comic History	George Athan	712	Read Book	Edit Book	Delete Book

[Back to Genre List](#)

[About](#) [Privacy](#) [Help](#) [Contact](#)

Limitation

- The website supports uploading and reading the books only in pdf format currently.

```
C:\WINDOWS\system32\cmd.exe
2. Exit
Enter choice: 1

sending request to https://localhost:44305/api/Files

Genre Name: Education GenreID: 3
Genre Name: Comics GenreID: 8
Genre Name: Romantic GenreID: 9
Genre Name: SciFi GenreID: 10
Enter GenreId in which you wish to add book: 3

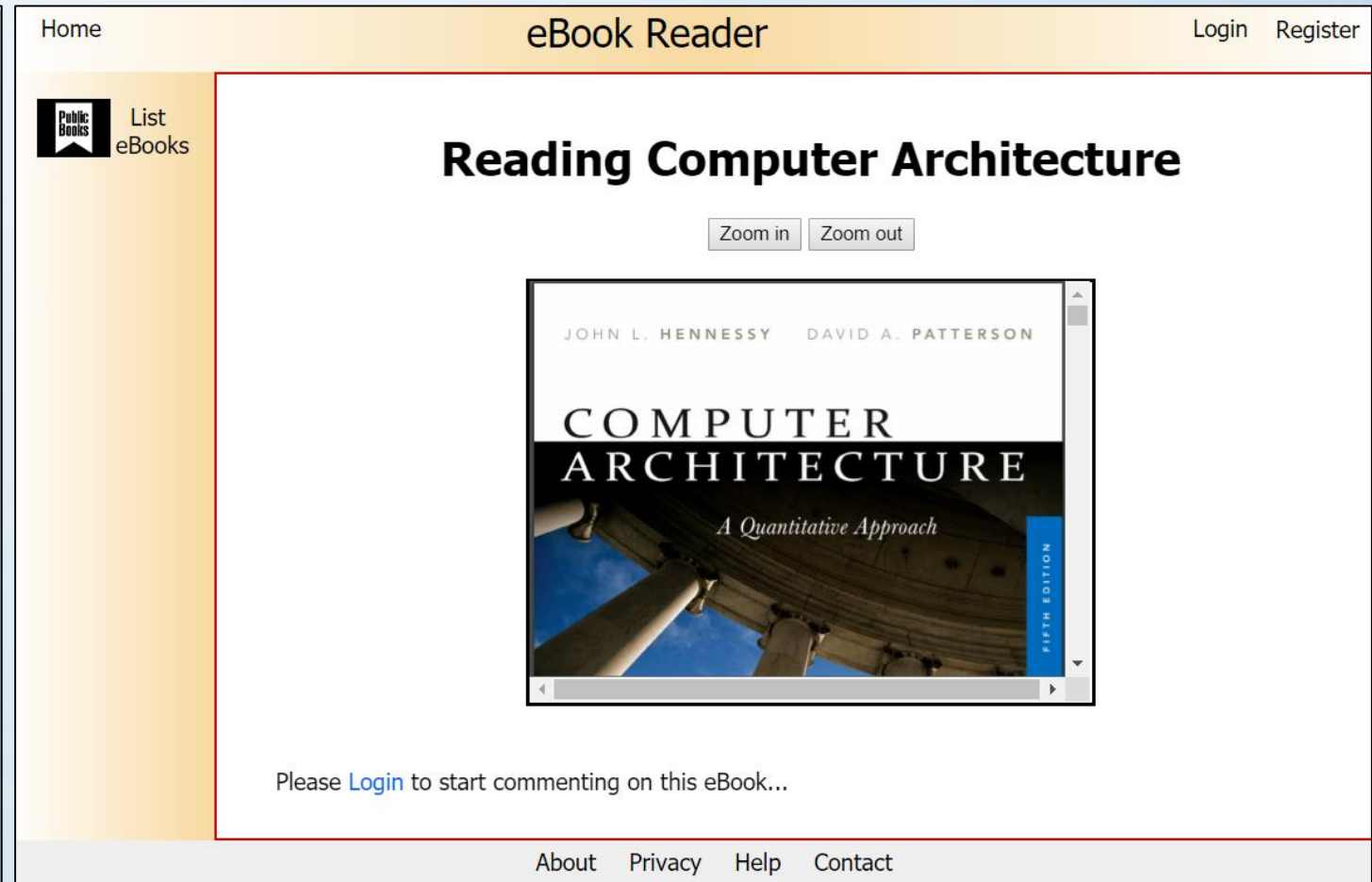
Enter book title: Computer Architecture

Enter book author: Patterson

Enter book ISBN: 34654

Book Access (Enter 1 for private, or 0 to keep book public: 0

Enter book path (only pdf files supported): C:\\Users\\Omkar\\source\\re
pos\\Project\\books\\ca.pdf
StatusCode: 200, ReasonPhrase: 'OK', Version: 1.1, Content: System.Net.H
ttp.HttpConnection+HttpResponseContent, Headers:
{
  Transfer-Encoding: chunked
  Server: Microsoft-IIS/10.0
  X-SourceFiles: =?UTF-8?B?QzpcVXNlcnNcT21rYXJcc291cmNlXHJlcG9zXFByb2p1Y
3RcV2ViQXBwbGljYXRpb25cYXBpXEXZpbGVz?=
  X-Powered-By: ASP.NET
  Date: Thu, 02 May 2019 06:31:09 GMT
}
Press 1 to continue, 0 to quit:
```





Thank You!