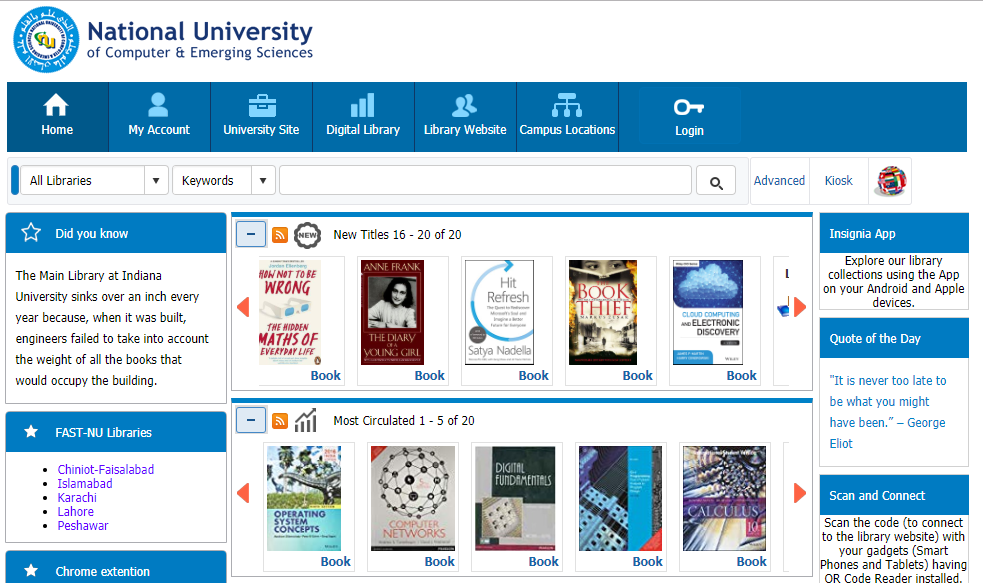
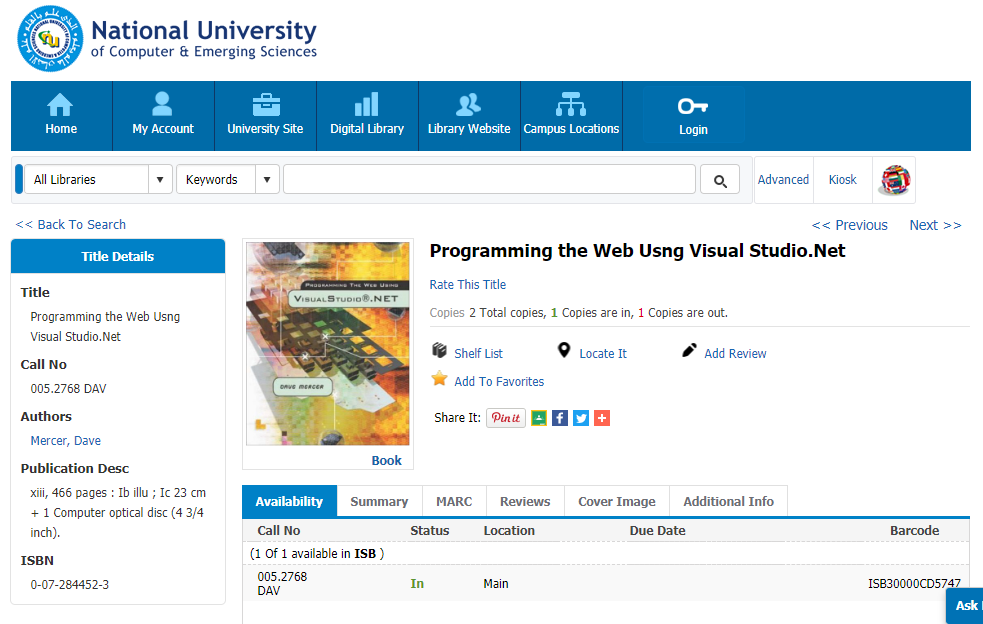
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **Web Programming** | **Course Code:** | **CS-406** |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Fall 2018** |
| **Duration:** | **60 Minutes** | **Total Marks:** | **40** |
| **Paper Date:** | **17-Nov-18** | **Weight** | **15 %** |
| **Section:** | **A/B** | **Page(s):** | **1 - 4** |
| **Exam:** | **Midterm-II** | **Reg. No.** |  |
| **Instruction/Notes:** | Do not bring any unauthorized material (e.g. written notes, notes in dictionaries, paper, sticky tape eraser etc.)  Solve the question on the answer sheets.  Good luck!  **SOLUTION** | | | |

Following is a snapshot of your university’s library home page

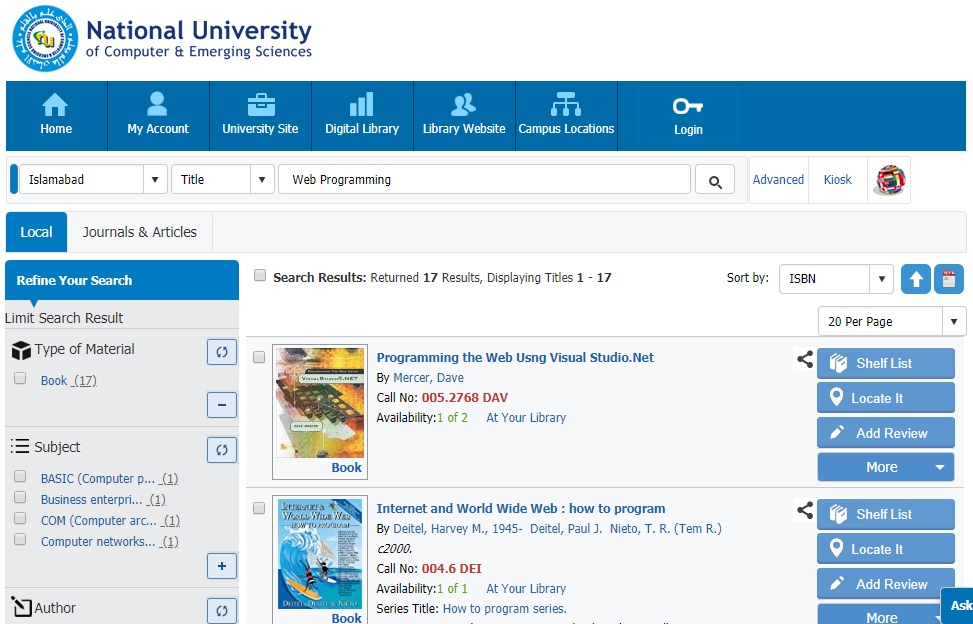
* The home page displays a list of new and most circulated book titles as follows:



* Clicking a book redirects the user to a new page which displays the book details as follows:



* Performing search by choosing location as “Islamabad” and search field as “Title” and entering the keywords “Web Programming” redirects the user to the search results page:



Your task is to work out the back-end architecture that supports these requests using the MVC pattern. The three requests are summarized as follows:

* <https://nu.insigniails.com/Library/Home>
* <https://nu.insigniails.com/Library/ItemDetail/52732>
* <https://nu.insigniails.com/Library/SearchResult>

1. List the MVC based folder structure of your application and mention the file names. (3 marks)

Controller: Home

Views: Home, ItemDetail, SearchResults.

Models: Books.cs, Library.cs , User.cs etc.

App Data: Database

1. Define the schema for the application (2 marks)

Books (id, name, author, title, call no, publication, available, available campus, barcode, most etc)

Library (id, name etc)

User (id, name, password, books etc)

1. Define the routes that handle the requests (5 marks)

app.UseMvc(routes =>

{

routes.MapRoute("Home ", " Home /{\* ItemDetail }",

defaults: new { controller = " Home ", action = " ItemDetail " });

routes.MapRoute("Home ", " Home /{\* SearchResults}",

defaults: new { controller = " Home ", action = " SearchResults" });

routes.MapRoute("default", "{controller=Home}/{action=Index}/{id?}");

});

1. Write the code for the middleware that processes these requests and sends appropriate response (30 marks)

namespace weconnect.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

//other details

Datacontext d = new Datacontext();

var q = SELECT MIN (x.most) from x in d.book where (x.id.equals(id)) select x;

return View(q.tolist());

}

public ActionResult ItemDetail(int id)

{

Datacontext d = new Datacontext();

var q = from x in d.book where (x.id.equals(id)) select x;

return View(q.tolist());

}

public ActionResult SearchResult(string n, string c)

{

Datacontext d = new Datacontext();

var q = from x in d.book where (x.name.equals(n) && x.availablecampus.equals(c)) select x;

return View(q.tolist());

}

}

Note: You don't have to write the code for the views but you have to name the files so that you can refer to those files in the middleware.

|  |
| --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** |