**E-Book**

A project report submitted as a partial fulfillment of the degree of

**Master of Computer Applications (MCA)**

**Submitted by**

## Bhuva Krupali Premjibhai

**Seat No. : MS21135**

# Of

**Post Graduate**

**Department of Computer science and Technology**

**Submitted to**

**SARDAR PATEL UNIVERSITY**

**Vallabh Vidyanagar**

**Post Graduate Department of Computer Science and Technology**

**Sardar Patel University**

**Vallabh Vidyanagar**

**CERTIFICATE**

This is to certify that **Bhuva Krupali Premjibhai**of MCA IV semester has worked on the project entitled “**E-Book**” satisfactory towards the partial fulfillment of the degree of Master of Computer Science and Applications during the final semester at the Post Graduate Department of Computer Science and Technology, Sardar Patel University, Vallabh Vidyanagar, Gujarat, India.

Date of Submission

1st May, 2023

**Internal Project Guide Head of the Department**

**Company certificate**

**Acknowledgement**

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, We rarely find words to express our gratitude towards those who were constantly involved with us.

The completion of any inter disciplinary project depends upon coordination, cooperation and combined efforts of several resources of knowledge, creativity, skill, energy and time. The work being accomplished now, We feel our most sincere urge to recall and knowledge through these lines, trying our best to give full credit wherever it deserves.

We would like to thank our project guide **Dr.Prashant P. Pittalia** and Dean & Principal **Dr. Priti Sajja** who advised and gave us moral support through the duration of our project. Without their constant encouragement we could not have been able to achieve what we have.

It’s our good fortune that we had support and well wishes of many. We are thankful to all and those names which have been forgotten to acknowledge here but contributions have not gone unnoticed.

With Sincere Regards,

Bhuva Krupali (MS21135)

**HISTORY OF DEPARTMENT**



The Department of Computer Science was established in the year 1986 as an institution offering postgraduate courses in Computer Science. The University started one-year Post Graduate Diploma in Computer Applications (PGDCA) course in the academic year 1975, Master of Computer Applications (MCA) in the year 1986, M.SC (Computer Science) in 1988, M. Sc. (Bioinformatics) in 2004, M. Phil. (Computer Science) in the year 2008, Ph. D. (Computer Science) In 1990 and Ph. D. (Bioinformatics) in 2011.

In the year 1988, Late Shree Gordhanbhai Hathibhai Patel donated money to the University for a Separate building of the Post Graduate Department of Computer Science. Several PC based systems had been installed and major research work was carried out in the areas of CAI (Computer Aided Instruction), KBS (Knowledge Based Systems) and DSS (Decision Support Systems). Three faculty members were selected on the basis of merit for the CICC-Japan training in the years 1992, 1994 and 1995. Early 2000 was the time when vast technological changes took place in the IT industry all over the world. The department could meet the technological requirements through support from the UGC, AICTE and various bodies of the State. A networked laboratory was established at the department, providing state of-the-art facilities. The department was recognized and offered ‘Refresher Courses in Computer Science’ for the Computer Science faculty members in India by the University Grants Commission. In the year 2002, a special MCA programme was introduced for providing lateral entry into the second year to computer science graduates. In a short span, the department grew to the extent that a new building was required. With one more generous donation from Late Shree Gordhanbhai Hathibhai Patel, grants from the UGC and the state government and the department’s internal sources, a new building of the department with modern infrastructure was constructed.

**Company Profile**



**PLUSINFOSYS**

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** |  | **Subject** | **Page Number** |
| 1 |  | **Project Profile** | 12 |
| 2 |  | **Introduction to tools** | 13 |
| 3 |  | **System Study** | 25 |
|  | 1 | Existing System | 26 |
|  | 2 | Proposed System | 26 |
|  | 3 | Scope of the Proposed System | 26 |
|  | 4 | Aim and Objective of the Proposed System | 26 |
| 4 |  | Modules | 29 |
| 5 |  | Feasibility Study | 32 |
|  | 1 | Operational Feasibility | 33 |
|  | 2 | Technical Feasibility | 33 |
|  | 3 | Economical Feasibility | 33 |
| 6 |  | **System Design** | 34 |
|  | 1 | Use Case Diagram | 35 |
|  | 2 | Activity Diagram | 37 |
|  | 3 | Class Daigram | 39 |
| 7 |  | Data Dictionary | 42 |
| 8 |  | Screen Layouts | 47 |
| 9 |  | Testing | 70 |
|  | 1 | Whitebox testing | 70 |
|  | 2 | BlackBox testing | 71 |
| 10 |  | Future Enhancement | 72 |
| 11 |  | Bibliography/References | 73 |

**Introduction of project**



* An**eBook** (Electronic Book; E-Book; Digital Book)  is the digital version of a printed book, or a full length book text that is published or converted to digital format to be viewed on the computer, dedicated ebook readers and mobile phones.
* An **ebook** (short for **electronic book**), also known as an **e-book** or **eBook**, is a [book](https://en.wikipedia.org/wiki/Book) publication made available in [digital](https://en.wikipedia.org/wiki/Digital_data) form, consisting of text, images, or both, readable on the [flat-panel display](https://en.wikipedia.org/wiki/Flat-panel_display) of computers or other electronic devices.Although sometimes defined as "an electronic version of a printed book", some e-books exist without a printed equivalent.
* E-books can be read on dedicated [e-reader](https://en.wikipedia.org/wiki/E-reader) devices, but also on any computer device that features a controllable viewing screen, including [desktop computers](https://en.wikipedia.org/wiki/Desktop_computer), [laptops](https://en.wikipedia.org/wiki/Laptop), [tablets](https://en.wikipedia.org/wiki/Tablet_computer) and [smartphones](https://en.wikipedia.org/wiki/Smartphone).
* In the 2000s, there was a trend of print and e-book sales moving to the [Internet](https://en.wikipedia.org/wiki/Internet" \o "Internet),where readers buy traditional paper books and e-books on [websites](https://en.wikipedia.org/wiki/Website) using [e-commerce](https://en.wikipedia.org/wiki/E-commerce) systems. With print books, readers are increasingly browsing through [images](https://en.wikipedia.org/wiki/Image) of the covers of books on publisher or bookstore websites and selecting and ordering titles online.

The paper books are then delivered to the reader by mail or another delivery service. With e-books, users can browse through titles online, and then when they select and order titles, the e-book can be sent to them online or the user can download the e-book. By the early 2010s, e-books had begun to overtake hardcover by overall publication figures in the U.S.

The main reasons for people buying e-books are possibly lower prices, increased comfort (as they can buy from home or on the go with mobile devices) and a larger selection of titles With e-books, "electronic [bookmarks](https://en.wikipedia.org/wiki/Bookmark_(World_Wide_Web)) make referencing easier, and e-book readers may allow the user to annotate pages."

"Although fiction and non-fiction books come in e-book formats, technical material is especially suited for e-book delivery because it can be digitally searched" for keywords. In addition, for programming books, code examples can be copied.

In the U.S., the amount of e-book reading is increasing. By 2014, 28% of adults had read an e-book, compared to 23% in 2013. By 2014, 50% of American adults had an [e-reader](https://en.wikipedia.org/wiki/E-reader) or a [tablet,](https://en.wikipedia.org/wiki/Tablet_computer) compared to 30% owning such devices in 2013.

**Project Profile**

**Project Profile**

* **Project Description:** in this project we developed one web application which is online book shopping system.

Basically using this application you can purchase book.

|  |  |
| --- | --- |
| **Project Title :** | **E-Book** |
| Company : | Plus Infosys |
| Project Guide : | Kinjal Bhatt |
| External Guide : | Dr Prashant P. Pittalia |
| Duration : | 4 Months. |
| Tools used: | SQL Server Management Studio, Visual Studio 2022 |
| Front End:: | HTML, CSS, BOOTSTRAP, AJAX, C# |
| Back End: | SQL Server, C#, ASP.NET Core |

**Team Information**

|  |  |  |
| --- | --- | --- |
| Name | Email-Id | Name Of Institute |
| Krupali Bhuva | krupali96@gmail.com | GDCST |
| Miti Halai | mitihalai@gmail.com | GVP |
| Riddhi Joshi | ridhhijoshi@gmail.com | GVP |

**Tools Used**

**Front End Tool**

**Bootstrap**



Bootstrap is a free and open-source CSS framework directed at responsive, mobile first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

**JavaScript**

****

• JavaScript is a lightweight, interpreted programming language.

• It is designed for creating network- centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

• JavaScript is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMA Script language specification.

• Alongside HTML and CSS, it is one of the three core technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern Web browsers without plug-ins.

• JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting objectoriented, imperative, and functional programming styles.

• It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as 25 networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

• Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics.

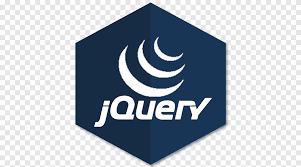
• The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by the self and Scheme programming languages.

• JavaScript is also used in environments that are not Webbased, such as PDF documents, site-specific browsers, and desktop widgets.

• Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation.

• It is also used in game development, the creation of desktop and mobile applications, and server-side network programming with runtime environments such as Node. JS.

**Jquery**



Jquery is a fast, small, and feature rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-touse API that works across a multitude of browsers. With a combination of versatility and extensibility, Jquery has changed the way that millions of people write JavaScript.

**HTML5**



• HTML stands for HYPER TEXT MARKUP LANGUAGE, which is most widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext.

• HTML was created by Berners - Lee in late 1991 but “HTML 2.0” was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999.

• Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

• As its name suggests, HTML is a Mark-up Language which means you use HTML to simply “mark-up” a text document with tags that tells a web browser how to structure it to display.

• Originally, HTML was develop with the intent of defining the structure of documents like heading, paragraph, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML.

**CSS3**



• Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a mark-up language Although most often used to set the visual style of web page and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media.

• Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

s

• CSS is designed primarily to enable the separation of document content from document presentation, 29 including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

• The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/ css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents

• CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

## **SQL Server**

## What is SQL Server?

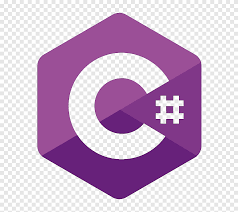
**SQL Server** is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database. SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

**T-SQL** is a Microsoft propriety Language known as **Transact-SQL.** It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

## Version History of SQL Server

* Microsoft and Sybase released version 1.0 in 1989.
* However, the partnership between these two ended in the early 1990s.
* Microsoft maintained ownership rights to the name SQL Server.
* Since the 1990s, subsequent versions of SQL Server have been released including SQL Server 2000, 2005, 2008, 2012, 2014, 2016, 2017, and 2019.

**C Sharp**

****

C# (pronounced "See Sharp") is a modern, object-oriented, and type-safe programming language. C# enables developers to build many types of secure and robust applications that run in .NET. C# has its roots in the C family of languages and will be immediately familiar to C, C++, Java, and JavaScript programmers.

This tour provides an overview of the major components of the language in C# 11 and earlier. If you want to explore the language through interactive examples, try the [introduction to C#](https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/) tutorials.

C# is an object-oriented, **component-oriented** programming language. C# provides language constructs to directly support these concepts, making C# a natural language in which to create and use software components.

Since its origin, C# has added features to support new workloads and emerging software design practices. At its core, C# is an **object-oriented** language. You define types and their behavior.

Several C# features help create robust and durable applications. [**Garbage collection**](https://learn.microsoft.com/en-us/dotnet/standard/garbage-collection/) automatically reclaims memory occupied by unreachable unused objects. [**Nullable types**](https://learn.microsoft.com/en-us/dotnet/csharp/nullable-references) guard against variables that don't refer to allocated objects. [**Exception handling**](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/) provides a structured and extensible approach to error detection and recovery.

[**Lambda expressions**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/lambda-expressions) support functional programming techniques. [**Language Integrated Query (LINQ)**](https://learn.microsoft.com/en-us/dotnet/csharp/linq/) syntax creates a common pattern for working with data from any source. Language support for [**asynchronous operations**](https://learn.microsoft.com/en-us/dotnet/csharp/asynchronous-programming/) provides syntax for building distributed systems.

C# has a [**unified type system**](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/). All C# types, including primitive types such as int and double, inherit from a single root object type.

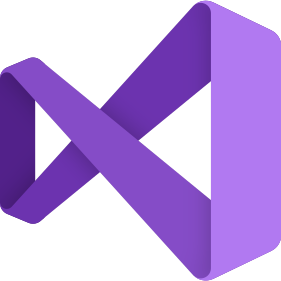
All types share a set of common operations. Values of any type can be stored, transported, and operated upon in a consistent manner. Furthermore, C# supports both user-defined [reference types](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/reference-types) and [value types](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/value-types). C# allows dynamic allocation of objects and in-line storage of lightweight structures. C# supports generic methods and types, which provide increased type safety and performance.

C# provides iterators, which enable implementers of collection classes to define custom behaviors for client code.

C# emphasizes **versioning** to ensure programs and libraries can evolve over time in a compatible manner.

Aspects of C#'s design that were directly influenced by versioning considerations include the separate virtual and override modifiers, the rules for method overload resolution, and support for explicit interface member declarations.

## **Microsoft .NET**



.NET is a free, cross-platform, open source developer platform for building many different types of applications.

With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, IoT, and more.

## **Languages**

You can write .NET apps in C#, F#, or Visual Basic.

* C# is a simple, modern, object-oriented, and type-safe programming language.
* F# is a programming language that makes it easy to write succinct, robust, and performant code.
* Visual Basic is an approachable language with a simple syntax for building type-safe, object-oriented apps.

## **Cross Platform**

Whether you're working in C#, F#, or Visual Basic, your code will run natively on any compatible operating system. You can build many types of apps with .NET. Some are cross-platform, and some target a specific set of operating systems and devices.

## **One consistent API**

.NET provides a standard set of base class libraries and APIs that are common to all .NET applications.

Each app model can also expose additional APIs that are specific to the operating systems it runs on, or the capabilities it provides.

For example, ASP.NET is the cross-platform web framework that provides additional APIs for building web apps that run on Linux or Windows.

**System StudyExisting System:**

Using Existing System User Can Purchase Book Form Online Store And Its Only Web Application.

* **Proposed System:**

Using Current Web Application Anyone Can Purchase Book From Application. Our Application is Platform Independent(Windows, IOS, Linux).Used Dependency Injection So We Can Reduce Duplication Of Code .

## **Admin Features of E-Book**

## **Dashboard** – For the admin dashboard, you will be able to all the basic access in the whole system. Such as summary of products, orders, and the categories

* **Manage Books**– The admin has access to the books management information system. He can add, update and delete the books.
* **Manage Categories** – The page where the admin can add, edit and delete categories information.
* **Manage Orders** – As the main functions of the admin, the admin can accept or reject the order from the customers on a case to case basis and the list of customer orders are listed.
* **Manage User**– The admin can manage the user’s account. Admin can add, update and Block user in the system.
* **Login and Logout** – By default one of the security features of this system is the secure login and logout system

## **Customer Features of eBook**

* **Login** **Page** – Customer enter their website credentials on this page to gain access in order to log in.
* **Register Page**– The page where new customer created their login credentials for the website.
* **Home Page**– When customer visit the website, this is the system’s default page. This page shows the books for sale in the store, or by entering a keyword in the search box above the books.
* **Book View Page** – The page on which the product’s specific information is shown, as well as the page on which the customer adds the product to his or her cart.
* **Cart List Page**– The page that lists the items that customer have chosen. This is the page where the customer can complete the order checkout process.
* **My Order Page** – The page that lists the customer’s orders.
* **Bash and Credit Card Payments** – This **Online eBook** o has a payment method that uses Credit Card Payments.
* **Aim and Objective of the Proposed System**

The main objective of the Online Book Store is to manage the details of Books, Customer, Payment, Delivery, Bills. It manages all the information about Books, Stock, Bills, Books.

The project is totally built at the administrative end and thus only the administrator is guaranteed access.

The purpose of the project is to build an application program to reduce the manual work for managing the Books, Customer, Stock, and Payment.

It tracks all the details about the Payment, Delivery, and Bills.

**Objective**

E-book is the interface between the students and librarian. It aims at improving the efficiency in the issue of books or magazines and reduces the complexities involved in it to the maximum possible extent

**Scope**

The system provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning).

The authority concerned with the issue of books can use this system to reduce his workload

**ModulesModule**

* **Admin**
* Can Login with valid credentials.
* Searching Books.
* See Books details.
* Add Books detail.
* Delete Books cards
* Manage[Edit Information] Books details.
* Manage his profile.
* **Users**
* Can register & login with valid credentials.
* Search category wise Books.
* See Books details.
* Manage cart.
* Place & Manage Orders.
* Manage his profile.
* Able to create new password.
* **Functional requirements:**
* It is defined as how they should react in the particular input and how the system should react in the particular situations and what the system do not do.
* In this project, login as functional requirement. In that functional requirement we may check the user name and password is correct or not. After checking entity of login, we can show the detail based on the type of actor.
* **Analysis:** In this place, the project requirement is analyzed and availability of requirement is seen.
* **Design:** Project manager makes the design of the project.
* **Implementation:** The construction of project is done and coding is developed.
* **Maintenance:** In this the software maintenance and the ways to avoid the drawback of software is made.

**System Analysis**

**Feasibility Study**

1. **Operational Feasibility**

* This is a very important aspect to be considered while developing a project.
* We decided the technology based on minimum possible cost factor. All hardware and software cost has to be borne by the organization.
* Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

**B. Technical Feasibility**

* This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system.
* For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SR), and checked if everything was Posible using different type of frontend and backend planforms.

**C. Economical Feasibility**

* No doubt the proposed system is fully GUl based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the b users so that they feel comfortable with new system.
* As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

**System Design**

**USE CASE DIAGRAMS**

**What is Unified Modeling Language (UML)?**

• The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the art facts of software systems, as well as for business modeling and other no-software systems.

• The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex system.

• The UML is a very important part of developing objects oriented software and the software development process.

• The UML uses mostly graphical notations to express the design of software projects.

• Using the UML helps project teams communicate, explore potential designs and validate the architectural design of the software.

**What is Goal of UML?**

The primary goals in the design of the UML were:

• Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.

• Provide extensibility and specialization mechanisms to extend the core concepts.

• Be independent of particular programming language and development process.

• Provide a formal basis for understanding the modeling language.

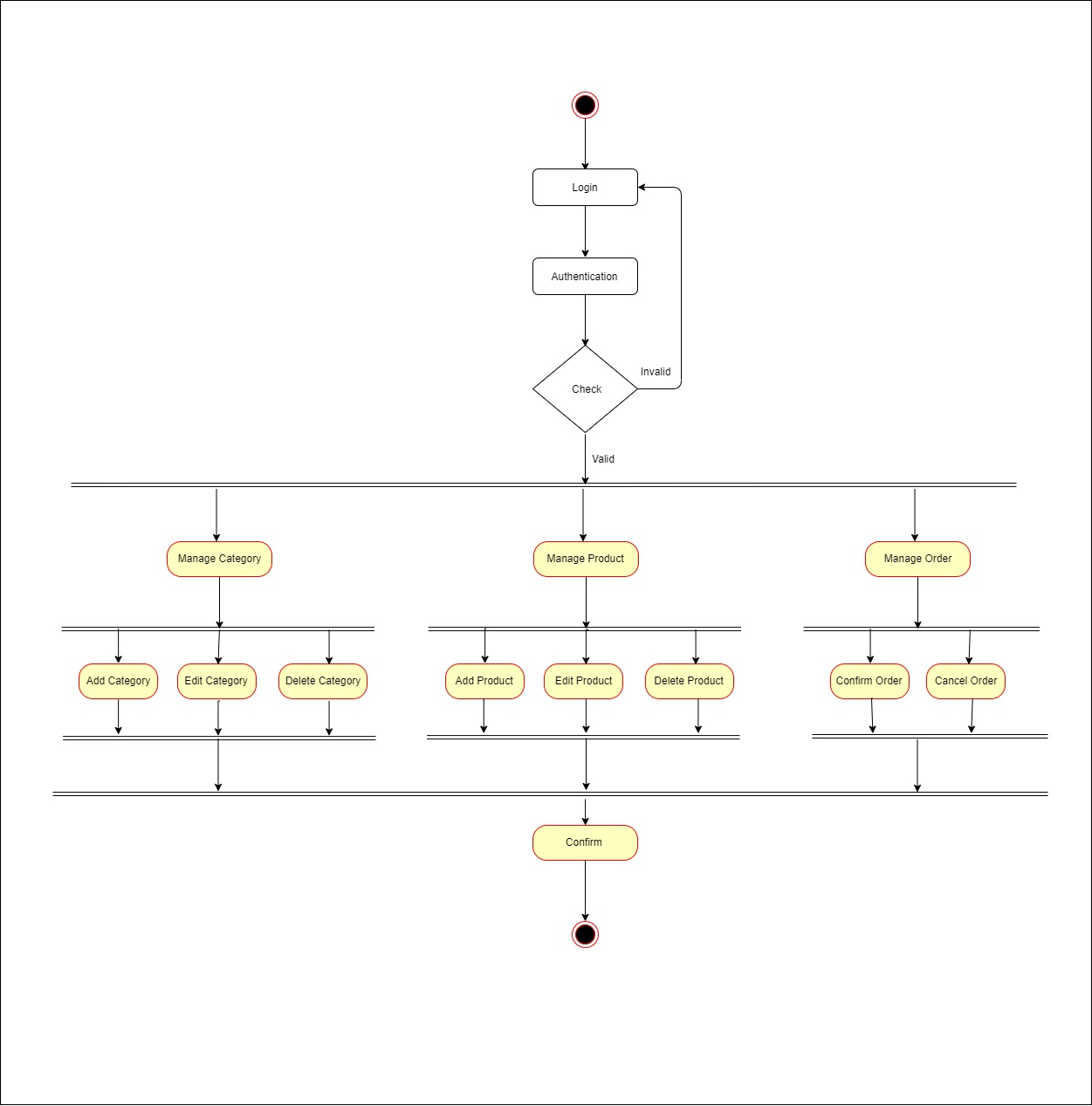
**Use Case Diagram:**



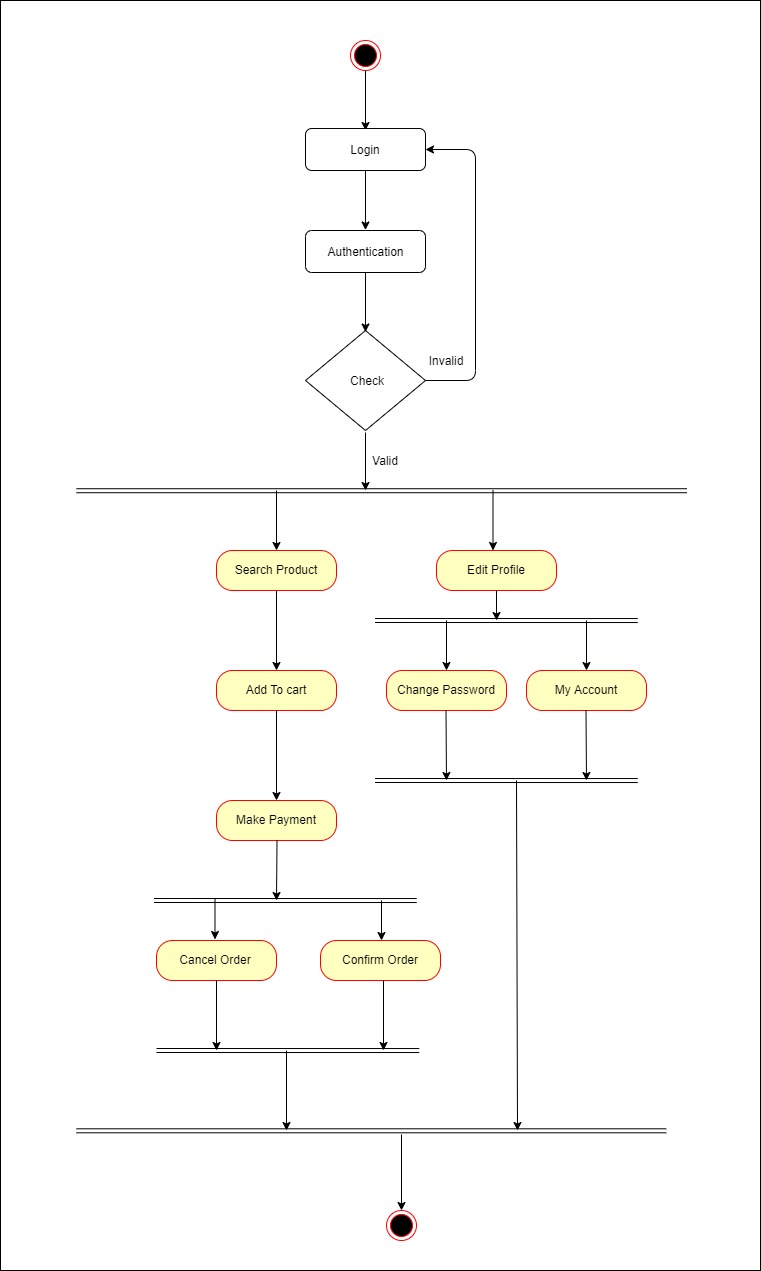
**Activity Diagram**

* The activity diagram used to describe flow of activity through a series of actions. Activity diagram is an important diagram to describe the system.
* The activity described as an action or operation of the system.

**User**



**Admin**



**CLASS DIAGRAM**

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The E-book Management system class diagram consists of five classes:

1. Login

2. Registered User

3. Administrator

4. Book

5. Visitor

6. Download

7. Logout

1) Login:

Login to the system

2) Registered User: It consists of six attributes and four operations. The attributes are user id, name, password, email id, phone no, security question. The operations of this class are download (), login (), search (), register ().

3) Administrator:

It consists of four attributes and two operations. The attributes are name, password, email id, admin id. The operations of this class are update (), record ().

4) Book:

It consists of four attributes and two operations. The attributes are book id, book name, author, and price. The operations of this class are update (), add ().

5) Visitor:

It consists of two attributes and two operations. The attributes are user name, email id. The operations of this class are search book(), read book().

6) Download:

It consists of two attributes and two operations. The attributes are user id, book id, date, and amount. The operations of this class are search download ().

7) Logout:

Logout from the system.

**CLASS DIAGRAM**



**Data Dictionary**Admin

**Primary Key :- Admin\_Id**

|  |  |  |
| --- | --- | --- |
| Field Name | DataType | Constraints |
| Admin\_Id | Int | NotNull |
| Name | Nvarchar(20) | Null |
| Address | Nvarchar(max) | Null |
| City | Nvarchar(20) | Null |
| Email | Nvarchar(max) | Null |
| Pincode | Nvarchar(10) | Null |
| State | Nvarchar(20) | Null |
| PhoneNumber | Nvarchar(15) | Null |
| Password | Nvarchar(18) | NotNull |

**Categories**

**Primary Key :- CategoryId**

|  |  |  |
| --- | --- | --- |
| Field Name | DataType | Constraints |
| CategoryId | Int | NotNull |
| Name | Nvarchar(20) | NotNull |
| CreatedDateTime | Datetime | NotNull |

User

**Primary Key:- UserId**

|  |  |  |
| --- | --- | --- |
| Field Name | DataType | Constraints |
| UserId | Int | NotNull |
| Name | Nvarchar(20) | Null |
| Address | Nvarchar(max) | Null |
| City | Nvarchar(20) | Null |
| Email | Nvarchar(max) | Null |
| Pincode | Nvarchar(10) | Null |
| State | Nvarchar(20) | Null |
| PhoneNumber | Nvarchar(15) | Null |
| Password | Nvarchar(18) | NotNull |

**Products**

Primary Key :- Product Id

Foreign Key :- CategoryId

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Constraints |
| ProductId | Int | NotNull |
| Name | Nvarchar(20) | NotNull |
| Description | Nvarchar(200) | NotNull |
| Price | Float | NotNull |
| Image | Nvarchar(Max) | NotNull |
| CategoryId | Int | NotNull |

**Order Table**

**Primary Key:- OrderId**

|  |  |  |
| --- | --- | --- |
| FeildName | DataType | Constraints |
| OrderId | Int | NotNull |
| UserId | Int | NotNull |
| DateOfOrder | DateTime | NotNull |
| DateOfShipping | DateTime | NotNull |
| Amount | Decimal | NotNull |
| OrderStatus | Nvarchar | Null |
| PaymentStatus | Nvarchar(50) | Null |
| TrackingNumber | Nvarchar(10) | Null |
| Carrier | Nvarchar(50) | Null |
| SessionId | Nvarchar(10) | Null |
| PaymentIntent | Nvarchar(5) | Null |
| DateOfPayment | DateTime | NotNull |
| DueDate | DateTime | NotNull |
| Name | Nvarchar(20) | NotNull |
| Phone | Nvarchar(15) | NotNull |
| Address | Nvarchar(50) | NotNull |
| City | Nvarchar(20) | NotNull |
| State | Nvarchar(20) | NOtNull |
| PostalCode | Nvarchar(10) | NotNull |

**Carts**

**Primary Key: - CartId**

**Foreign Key: - ProductId , UserId**

|  |  |  |
| --- | --- | --- |
| Field Name | DataType | Constraints |
| CartId | Int | NotNull |
| ProductId | Int | NotNull |
| Count | Int | NotNull |
| UserId | Int | NotNull |

Order Details

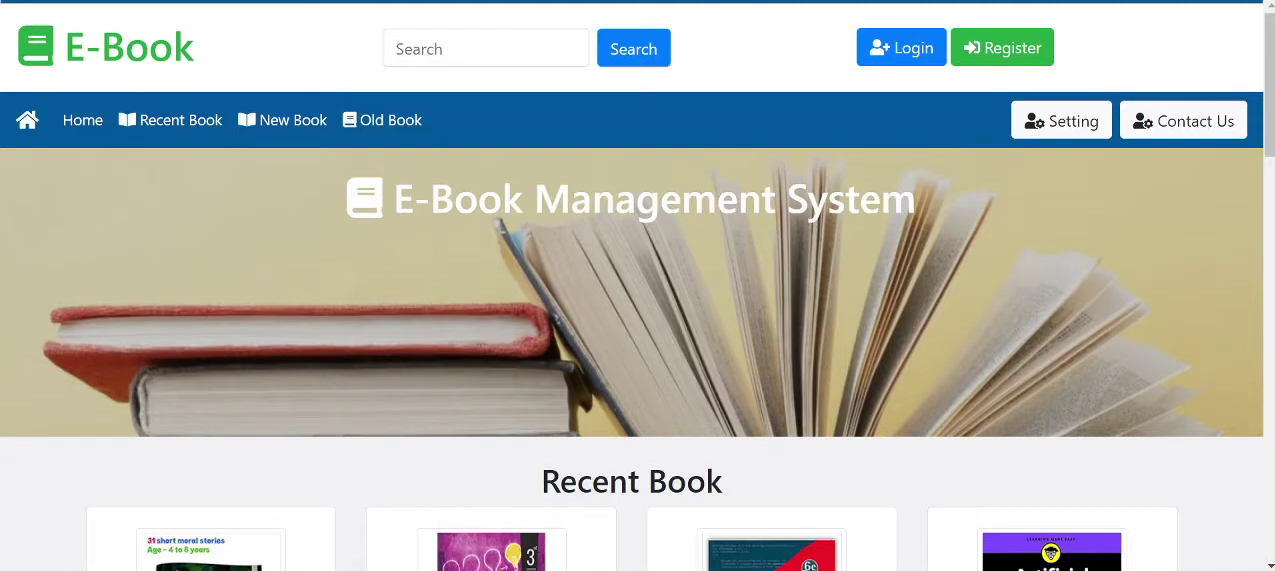
PrimaryKey:- Id

ForeignKey:- OrderId,ProductIdS

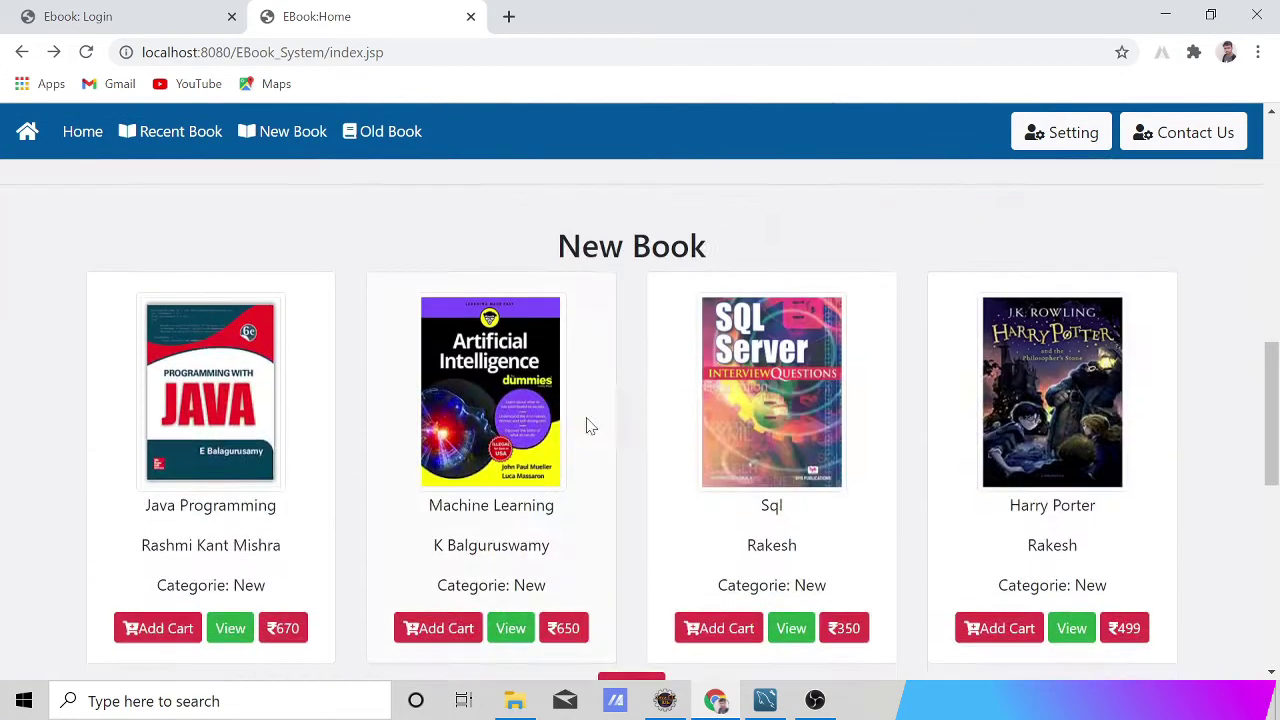
|  |  |  |
| --- | --- | --- |
| Field Name | DataType | Constraints |
| Id | Int | NotNull |
| OrderId | Int | NotNUll |
| ProductId | Int | NotNull |
| Price | Decimal | NotNull |
| Count | Int | NotNull |

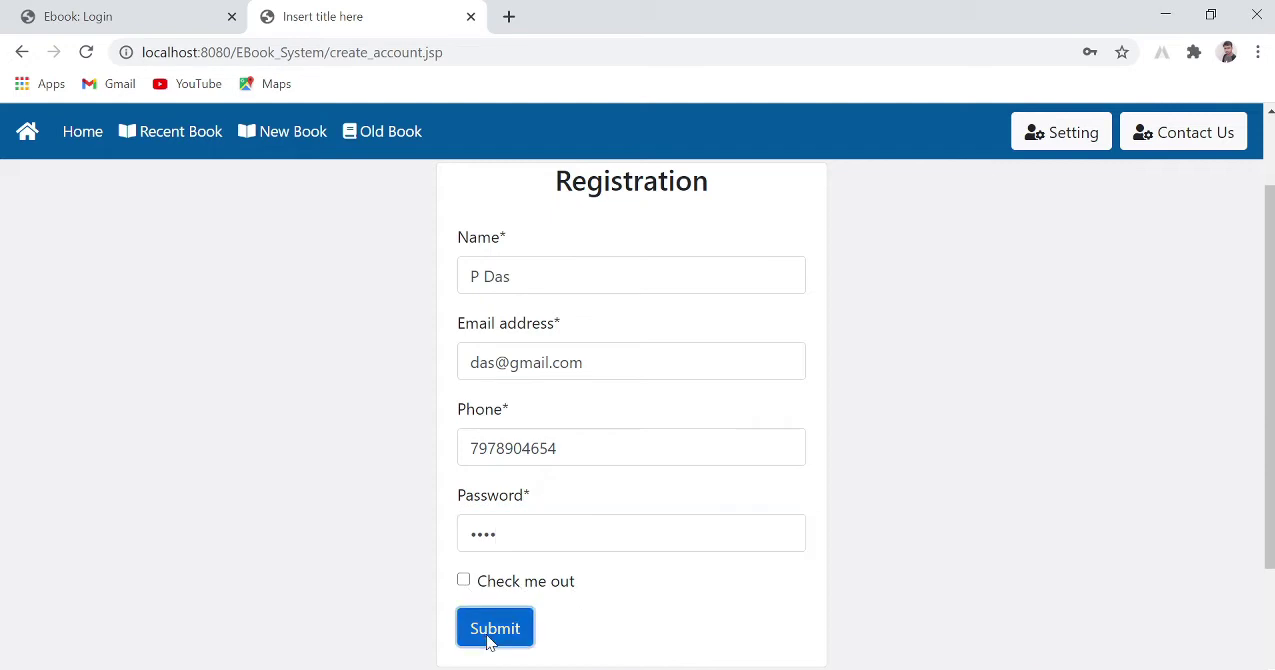
**Screen Layouts**

**Home Page**

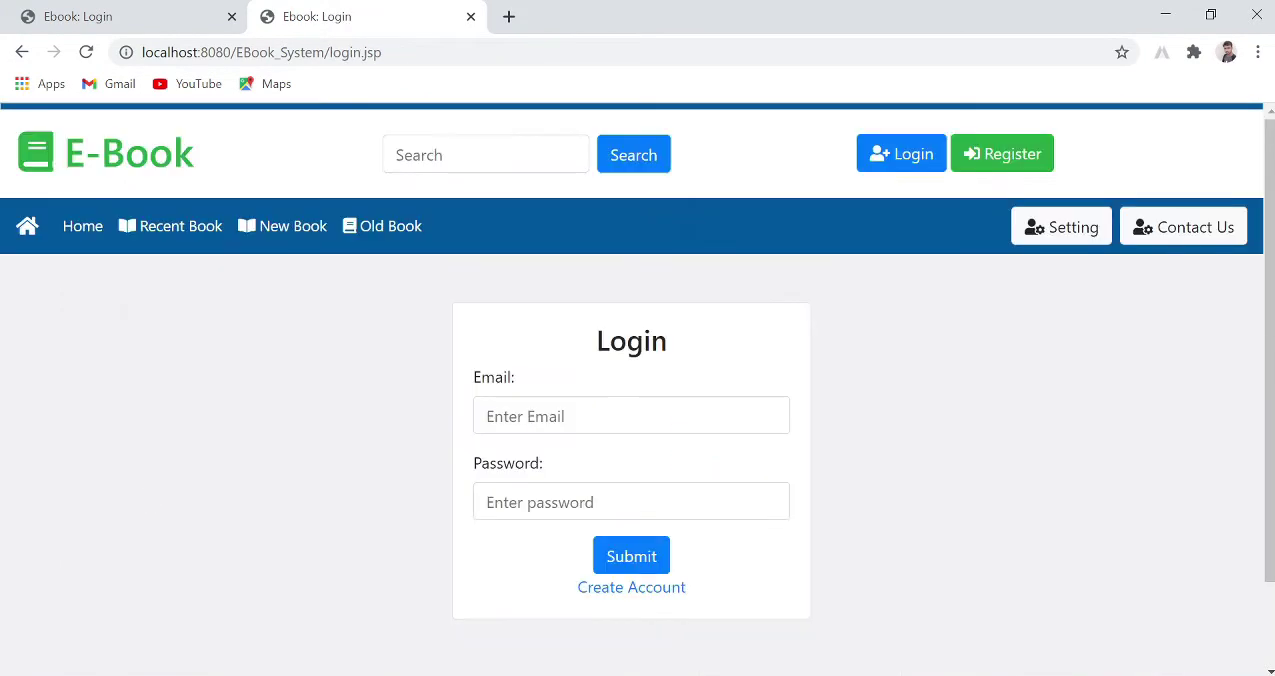
****

**Home Page**

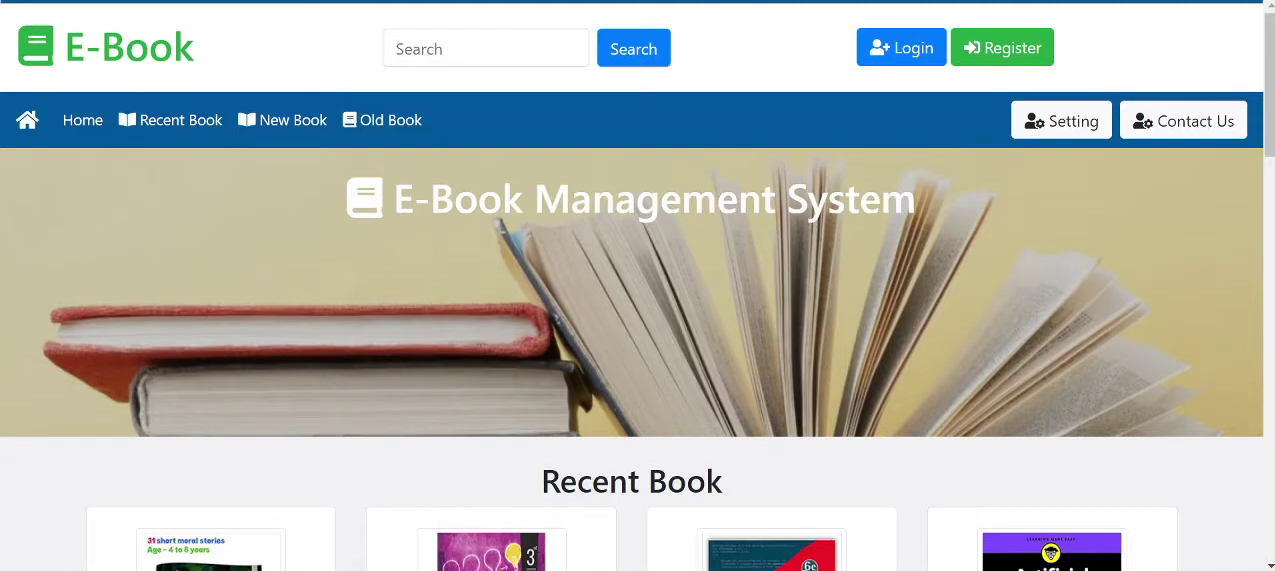


**Registration**

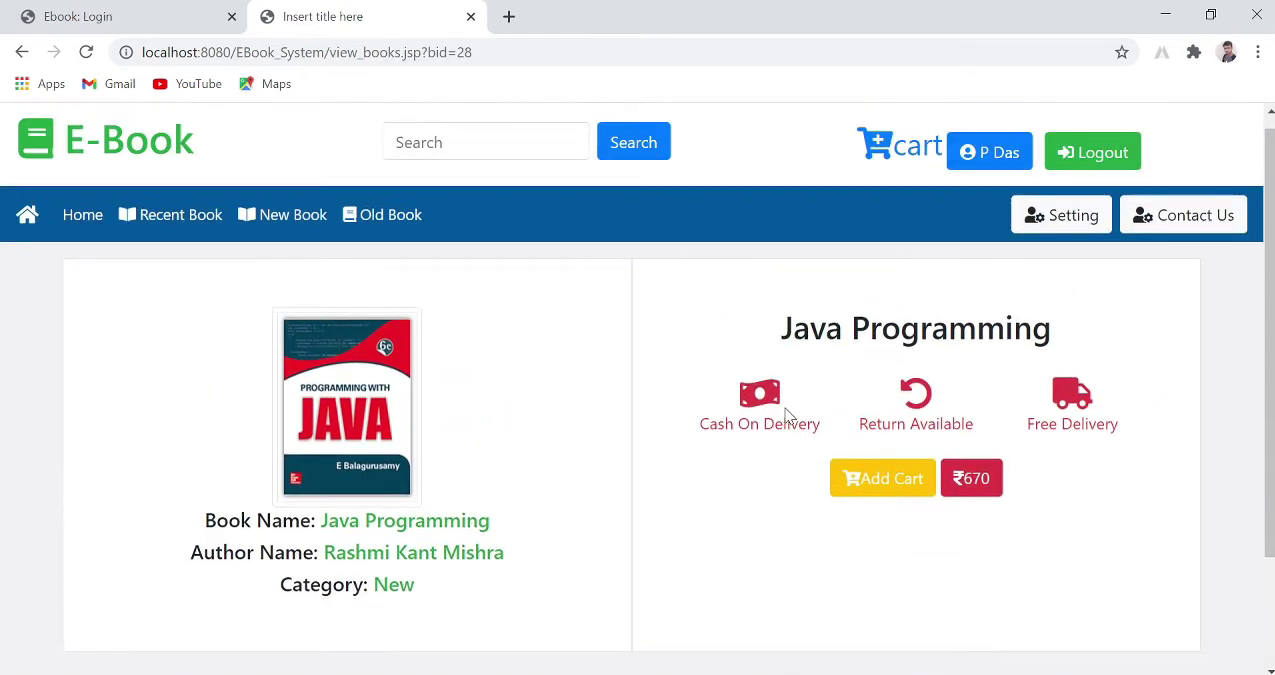
**Login Page**



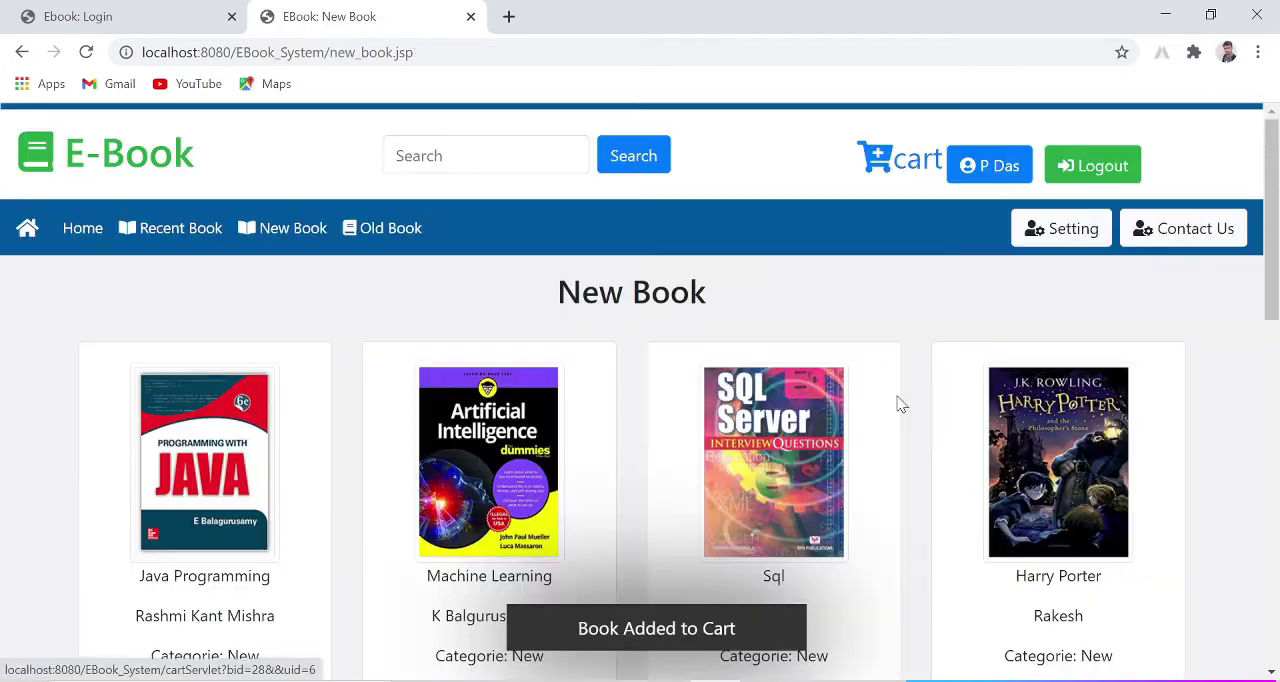
**Home Page**



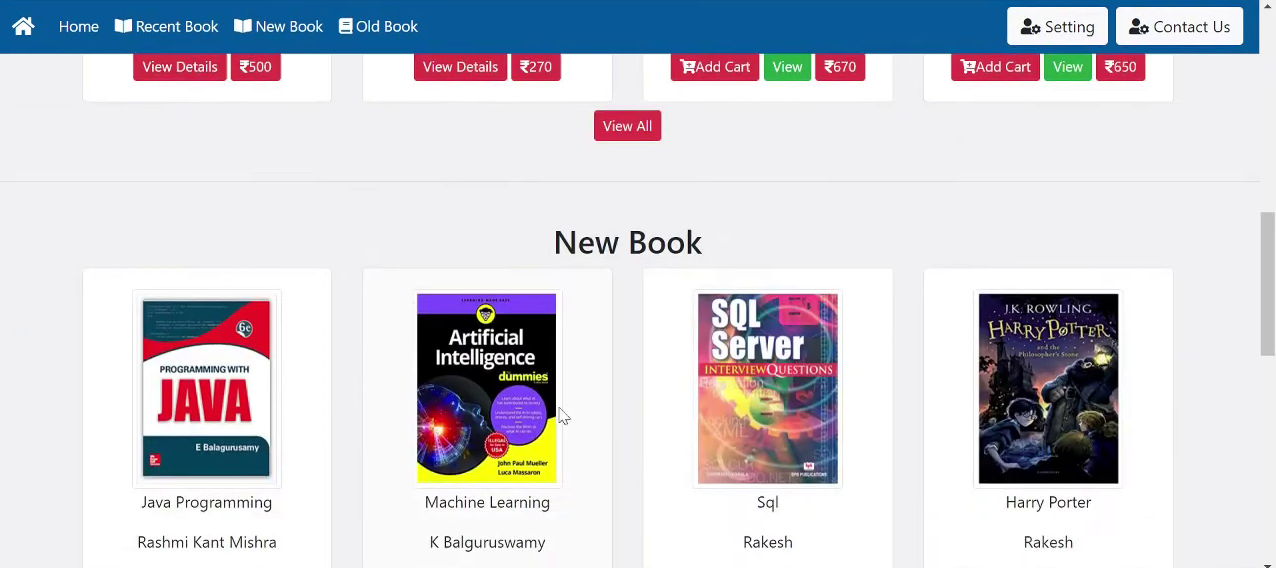
**View Book**



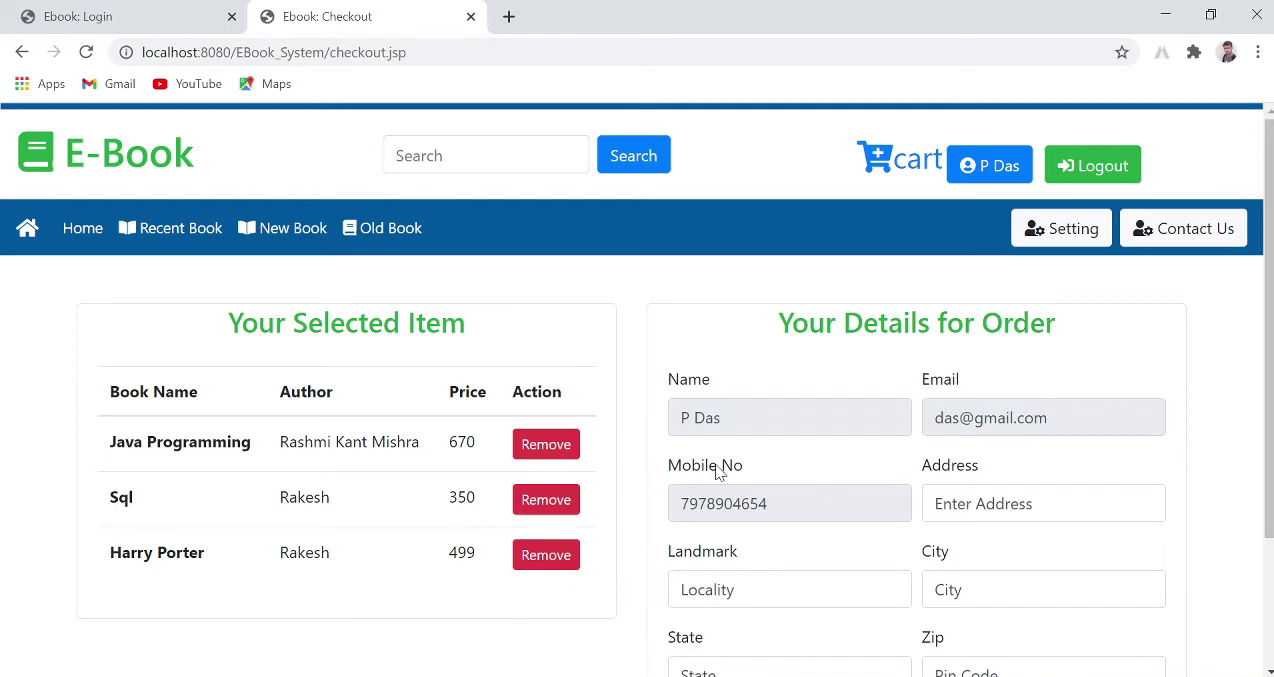
**Add to Cart**

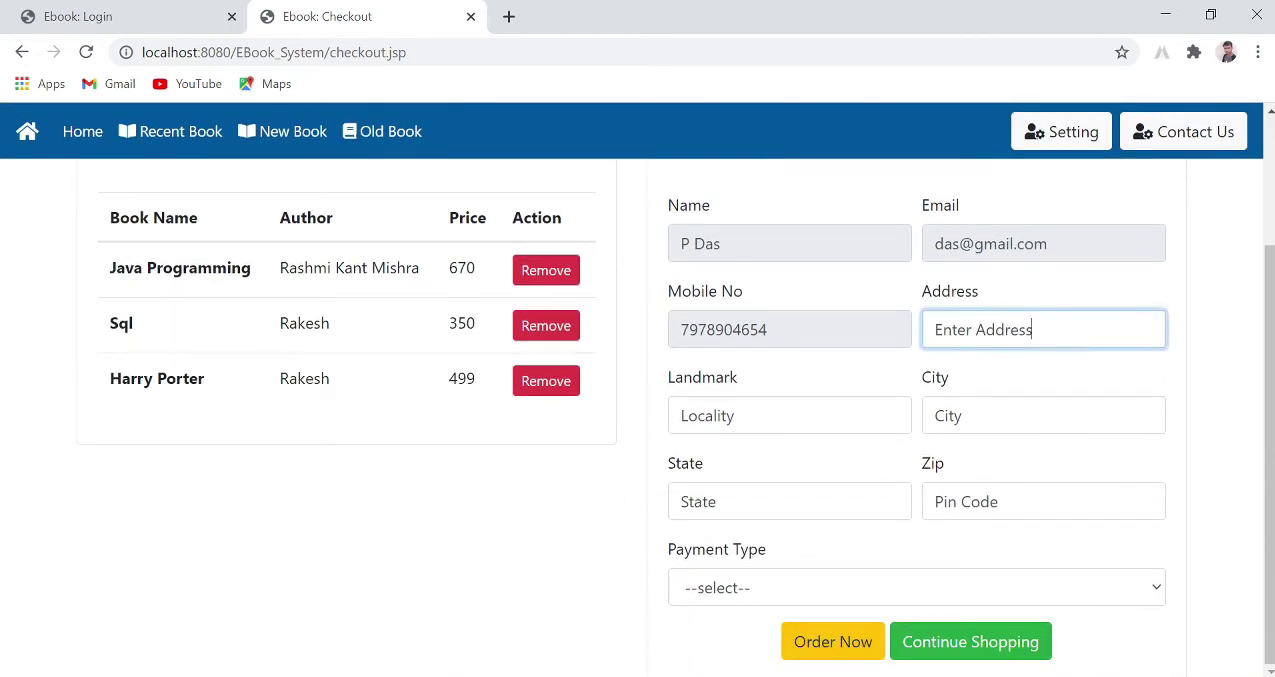


**Add to Cart**



**Order Details**

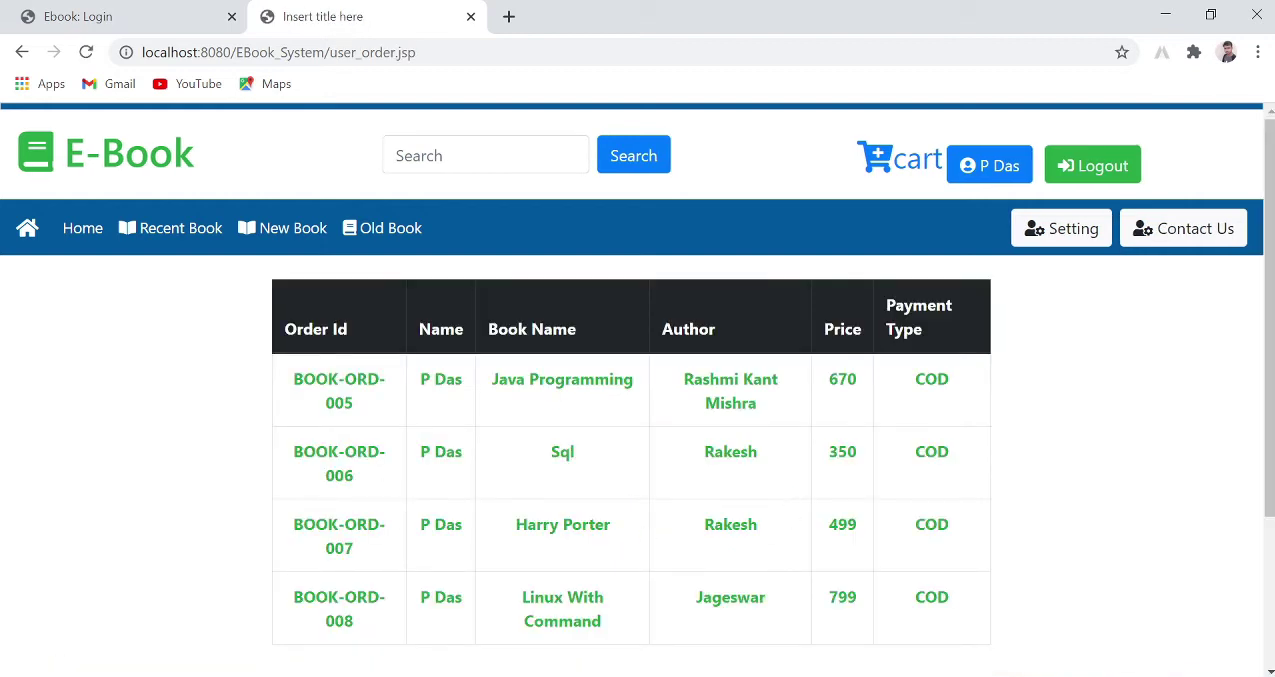


**Order Details**

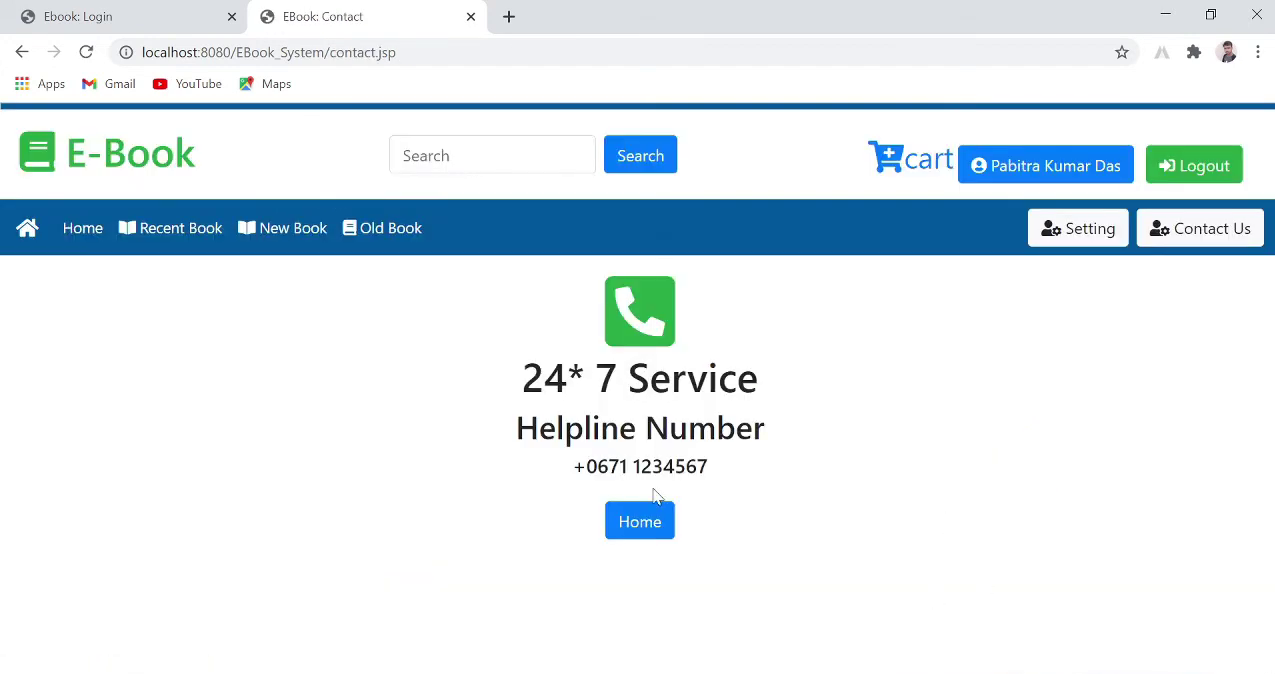
**Order Successfully**

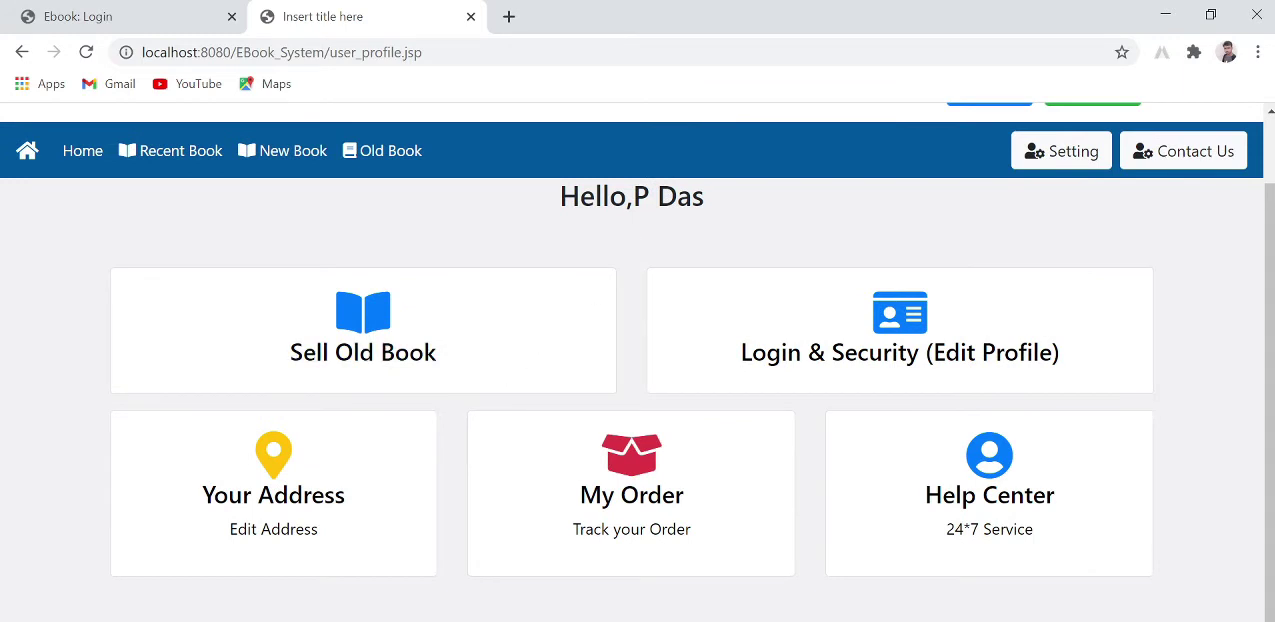


**View Order**

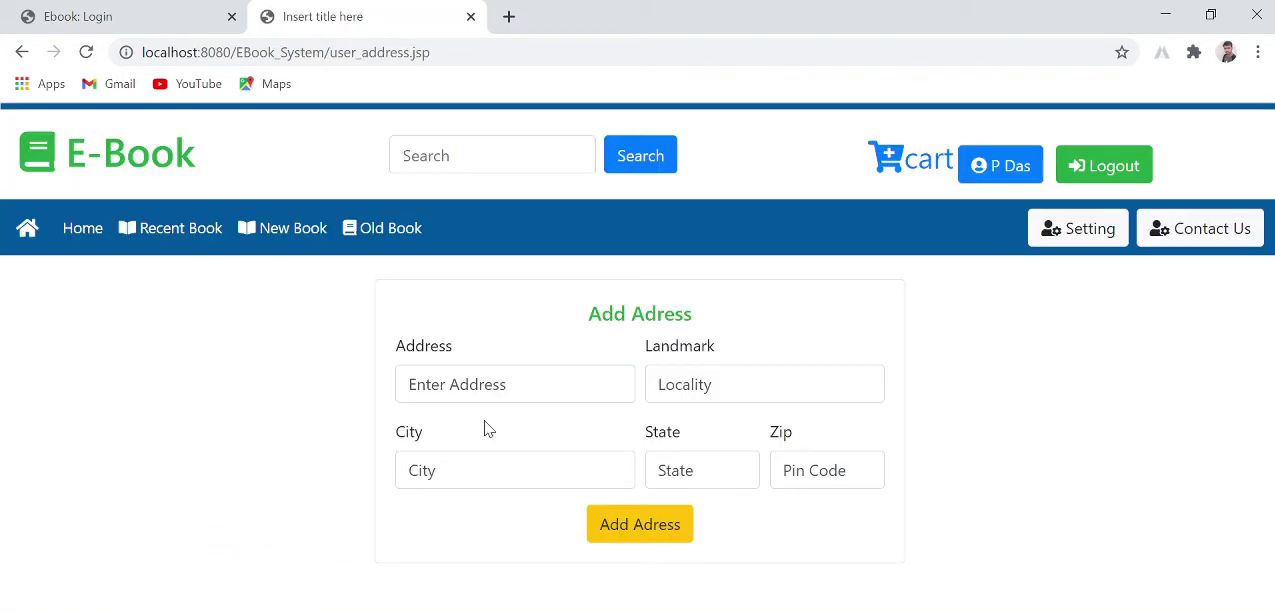


**Contact Us**

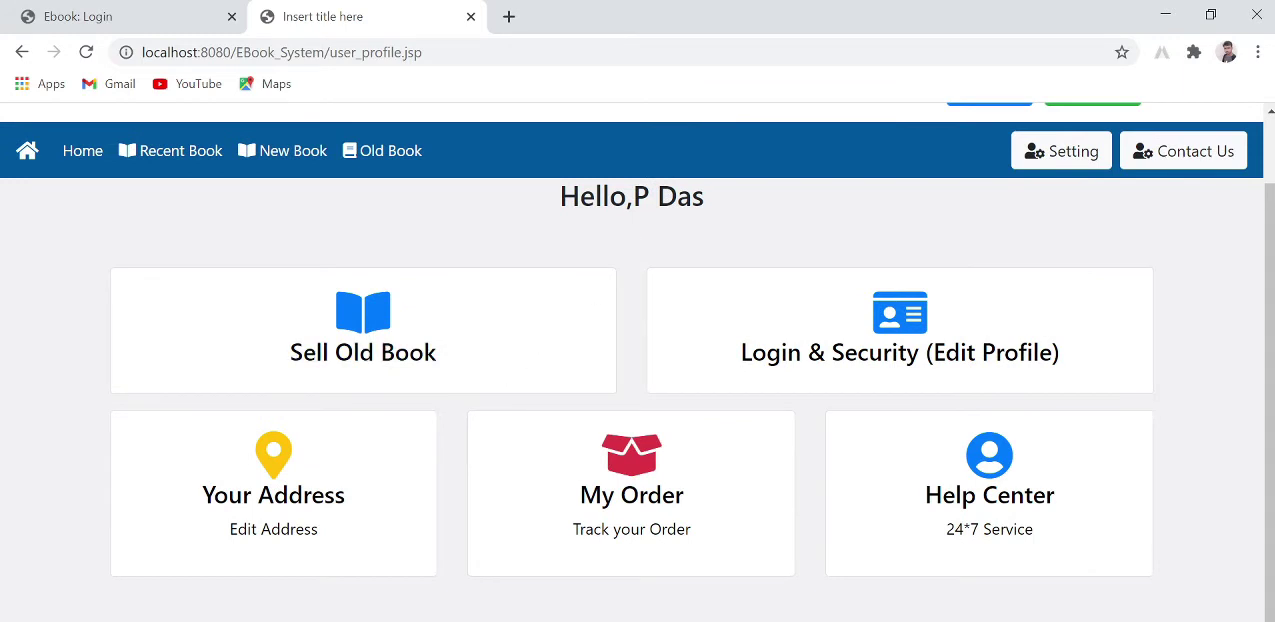


**Setting**

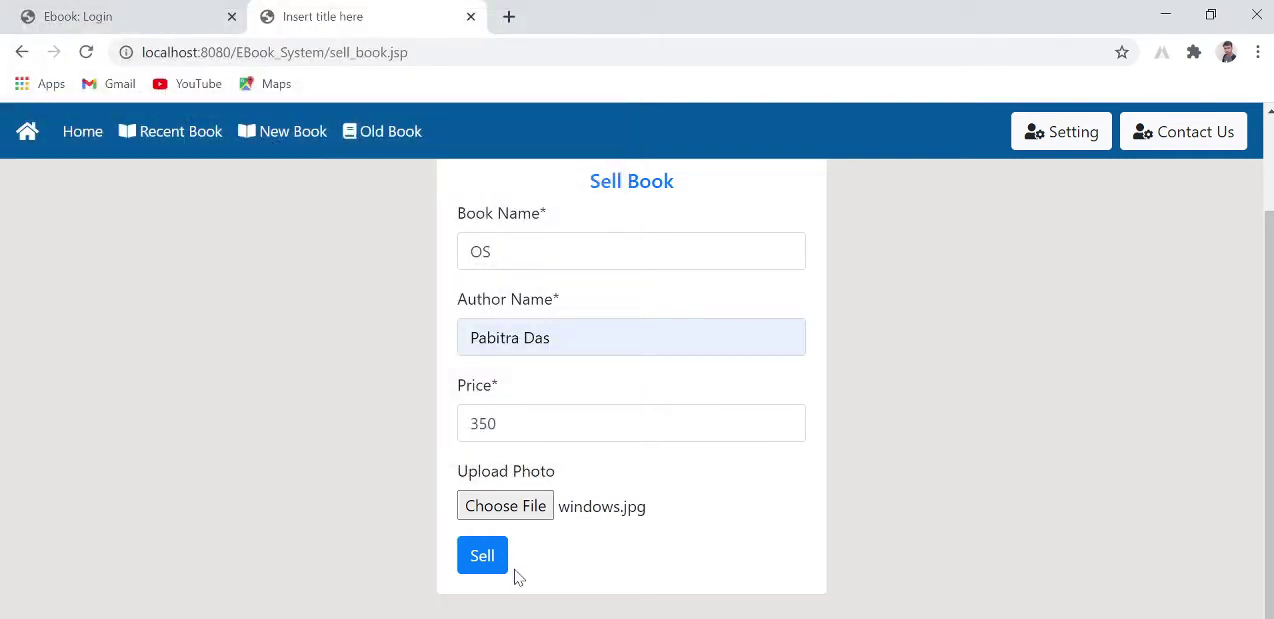
**Add Address**

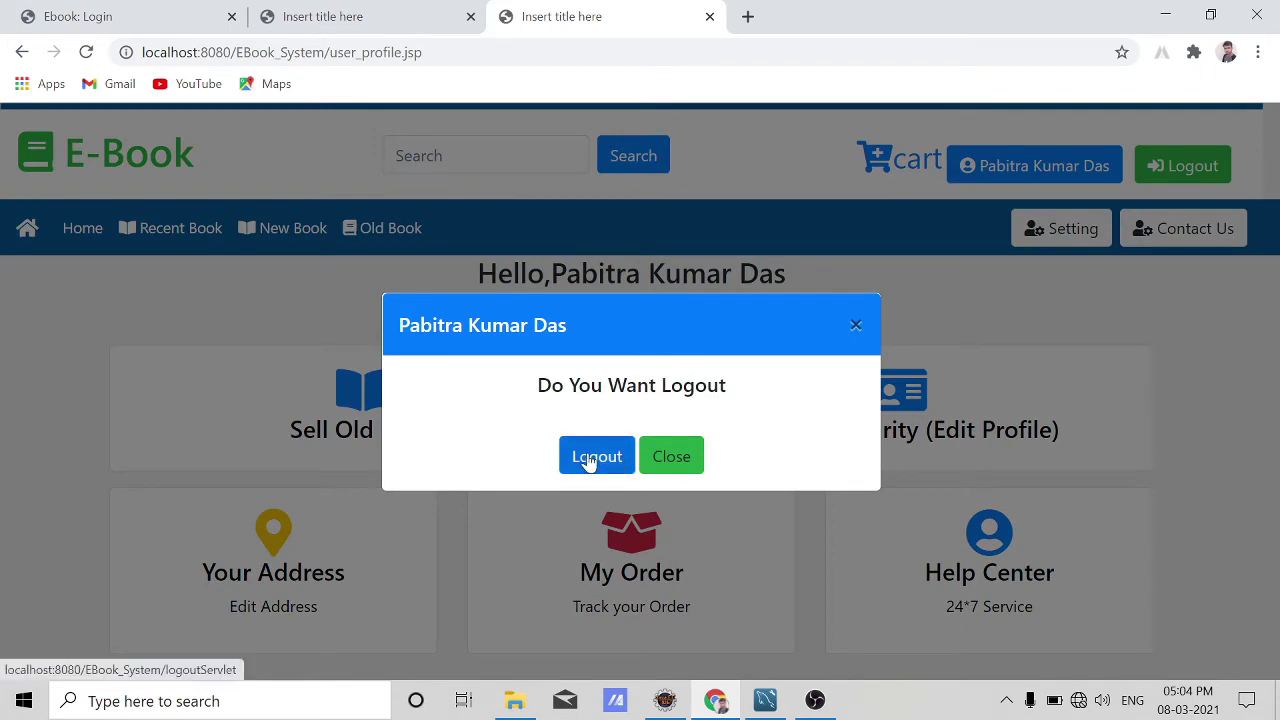


**Sell Book**

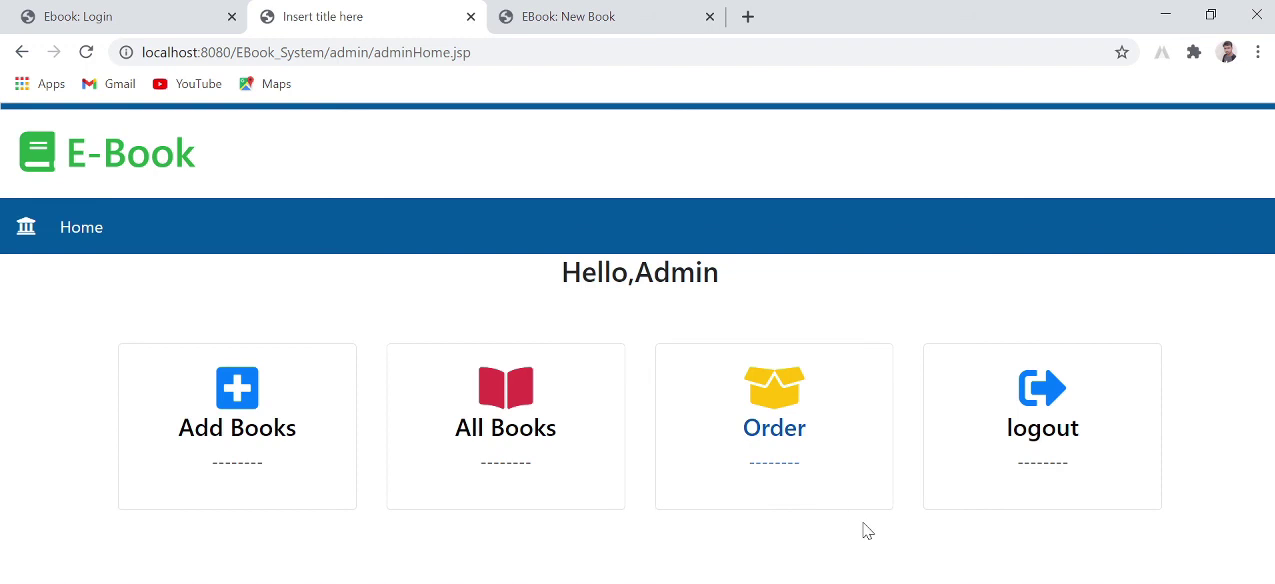


**Sell Old Book**

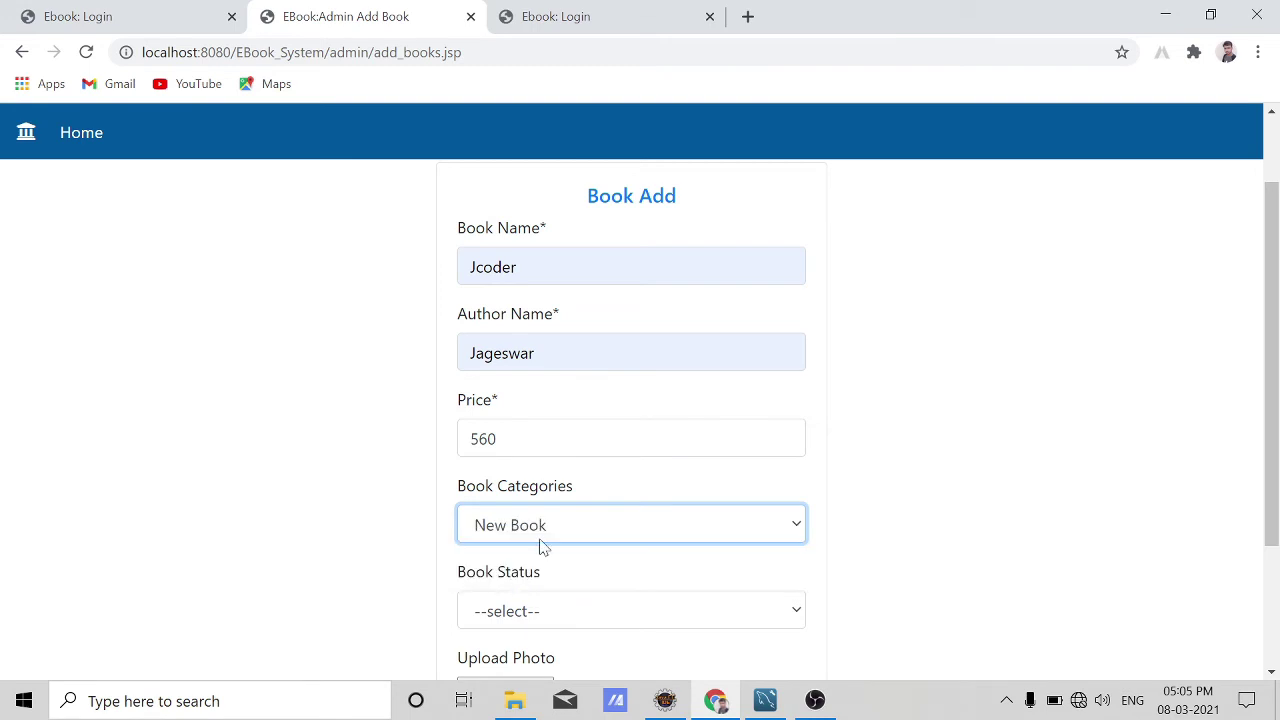


**Logout**

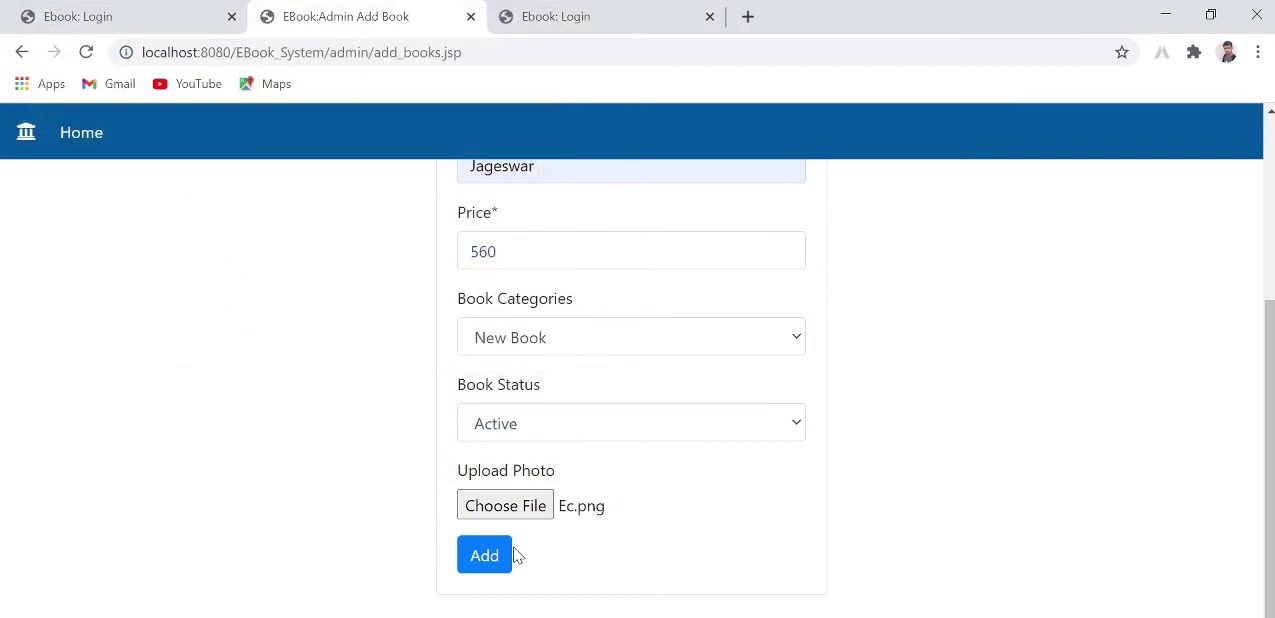
**Admin Dashboard**



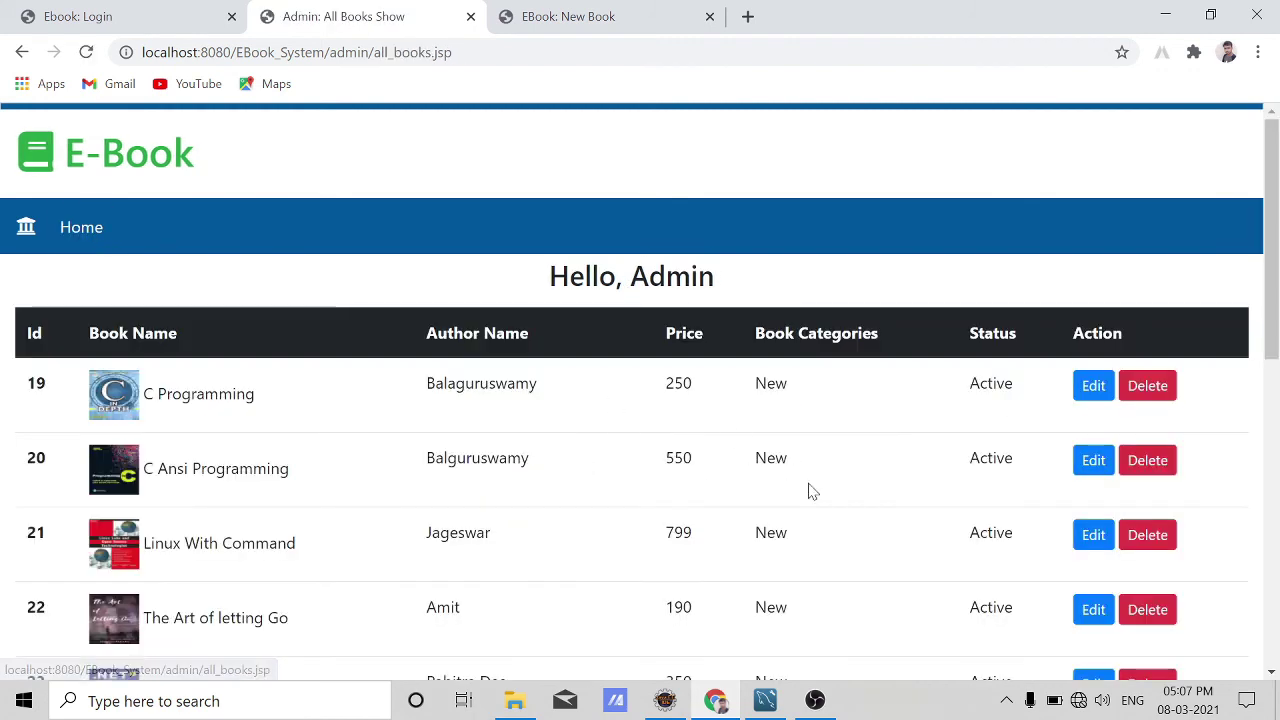
**Add Book**



**Add Book**



**All Book**



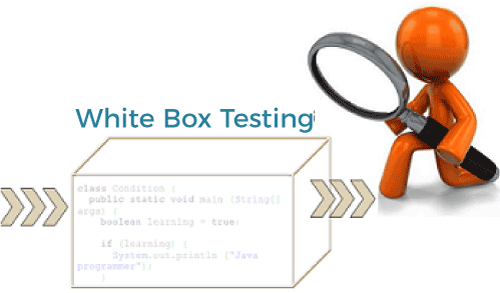
**Testing**

**1.white box**

**2.black box**

## **White Box testing**

The term 'white box' is used because of the internal perspective of the system. The **clear box or white box, or transparent box** name denotes the ability to see through the software's outer shell into its inner workings.



It is performed by Developers, and then the software will be sent to the testing team, where they perform black-box testing.

The main objective of white-box testing is to test the application's infrastructure. It is done at lower levels, as it includes unit testing and integration testing.

It requires programming knowledge, as it majorly focuses on code structure, paths, conditions, and branches of a program or software.

The primary goal of white-box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

It is also known as structural testing, clear box testing, code-based testing, and transparent testing. It is well suitable and recommended for algorithm testing.

## **Black Box testing**

The primary source of black-box testing is a specification of requirements that are stated by the customer. It is another type of manual testing.

It is a software testing technique that examines the functionality of the software without knowing its internal structure or coding.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function.

In this testing, the test engineer analyzes the software against requirements, identifies the defects or bugs, and sends it back to the development team.



In this method, the tester selects a function and gives input value to examine its functionality, and checks whether the function is giving the expected output or not. If the function produces the correct output, then it is passed in testing, otherwise failed.

Black box testing is less exhaustive than White Box and Grey Box testing methods. It is the least time-consuming process among all the testing processes. The main objective of implementing black box testing is to specify the business needs or the customer's requirements.

In other words, we can say that black box testing is a process of checking the functionality of an application as per the customer's requirement. Mainly, there are three types of black-box testing**: functional testing, Non-Functional testing,**and **Regression testing**. Its main objective is to specify the business needs or the customer's requirements.

Future Enhancement

While users acknowledge that adoption of eBooks will continue to increase, they do not envision print books disappearing within the near future.

Users expect that the transition to eBooks will happen fastest for research-related activities, rather than study, teaching, or leisure purposes.

They also expect that reference works will most quickly make the transition to eBooks, followed by research monographs and textbooks.

1. The user survey found that users most frequently locate eBooks through general search engines like Google as well as through online library catalogs

2. Users also said that they primarily use e-Books for research or study purposes, rather than for leisure or teaching purposes.

3. The types of eBooks most frequently used are reference works and textbooks.

4. Users value the convenience and ease of access that eBooks provide and are engaging in new forms of book content usage to take advantage of their libraries’ growing eBook collections.

5. Print books are perceived to have an advantage in ease and enjoy of reading 6. Finally, while e-Books will not replace print books in the near future, users are rapidly adopting them as complementary to print books.

6. An enhanced eBook typically includes all or some of the following elements:

* Hidden searchable text
* Collapsible table of contents
* Rich media features including audio, video, and interactivity
* Internal linking – footnotes/annotations/cross references/indexes
* External linking – geolocation/directions/associated websites/social media sites.

**Bibliography**

References

• <https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials>

• <https://www.c-sharpcorner.com/>

• <https://docs.microsoft.com/en-us/aspnet/mvc/>

• <https://www.tutlane.com/tutorial/aspnet-mvc>

• <https://www.w3schools.com/bootstrap/bootstrap_modal.asp>

• <https://jqueryui.com/>