A close up of a logo

Description generated with very high confidence

CMPS342 - Database Systems

Faculty of Science

Dept. of Mathematics and Computer Science

[**WWW.ONLINELIBRARYSYSTEM.SOMEE.COM**](http://www.onlinelibrarysystem.somee.com)

Online Library System

*Presented to:*

**Dr. Riham Abdel Kader**

*By:*

**Omar Mneimneh**

**Alaa Ghader**

**Karim Kayssi**

**Omar Orabi**

**Wassim Issa**

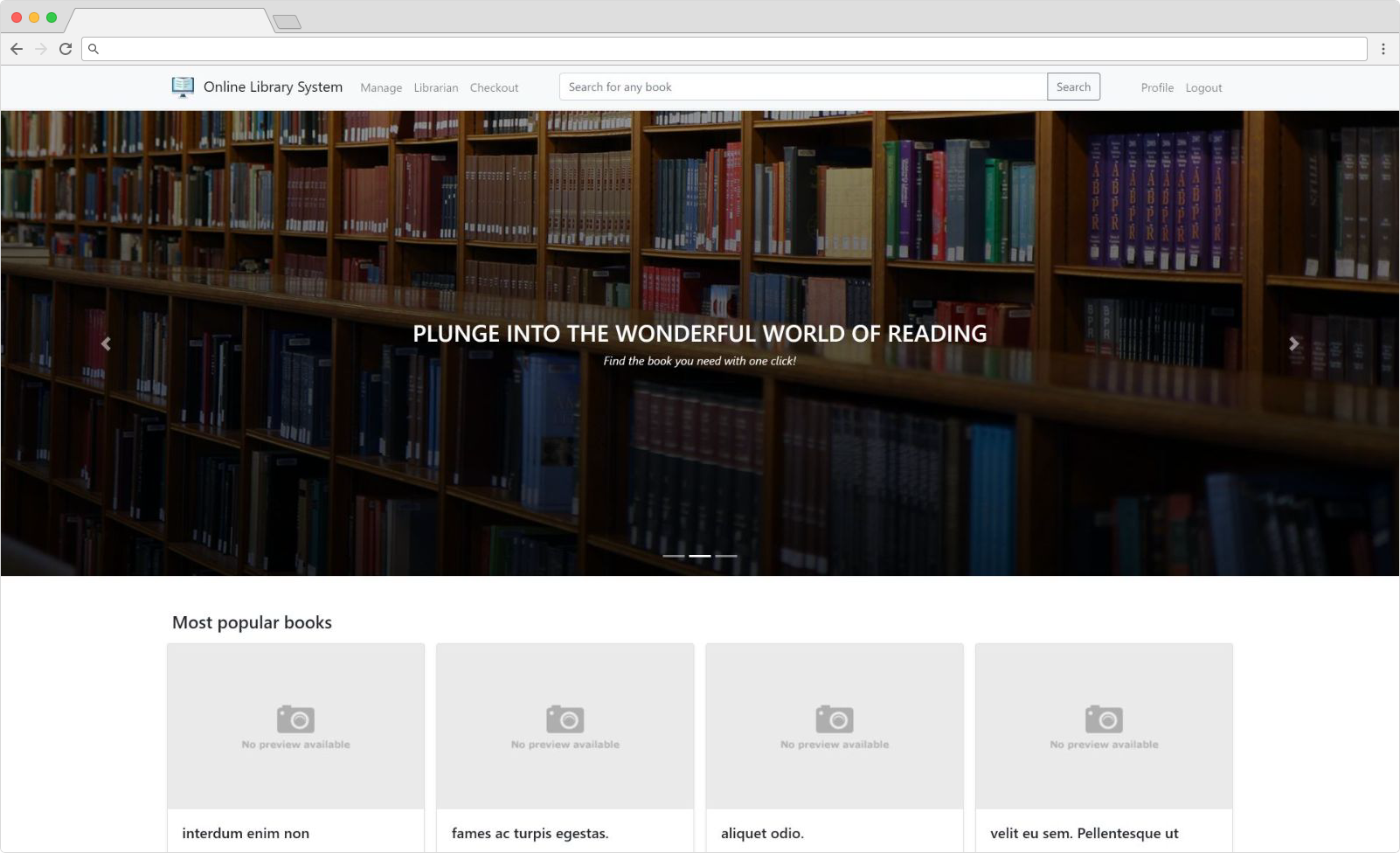
**Abdelrahman Daher**

# Pages Description

## Home Page

This page provides a nice welcome screen to the user, showing him some of the most popular books so he can interact and easily learn how to use the website.

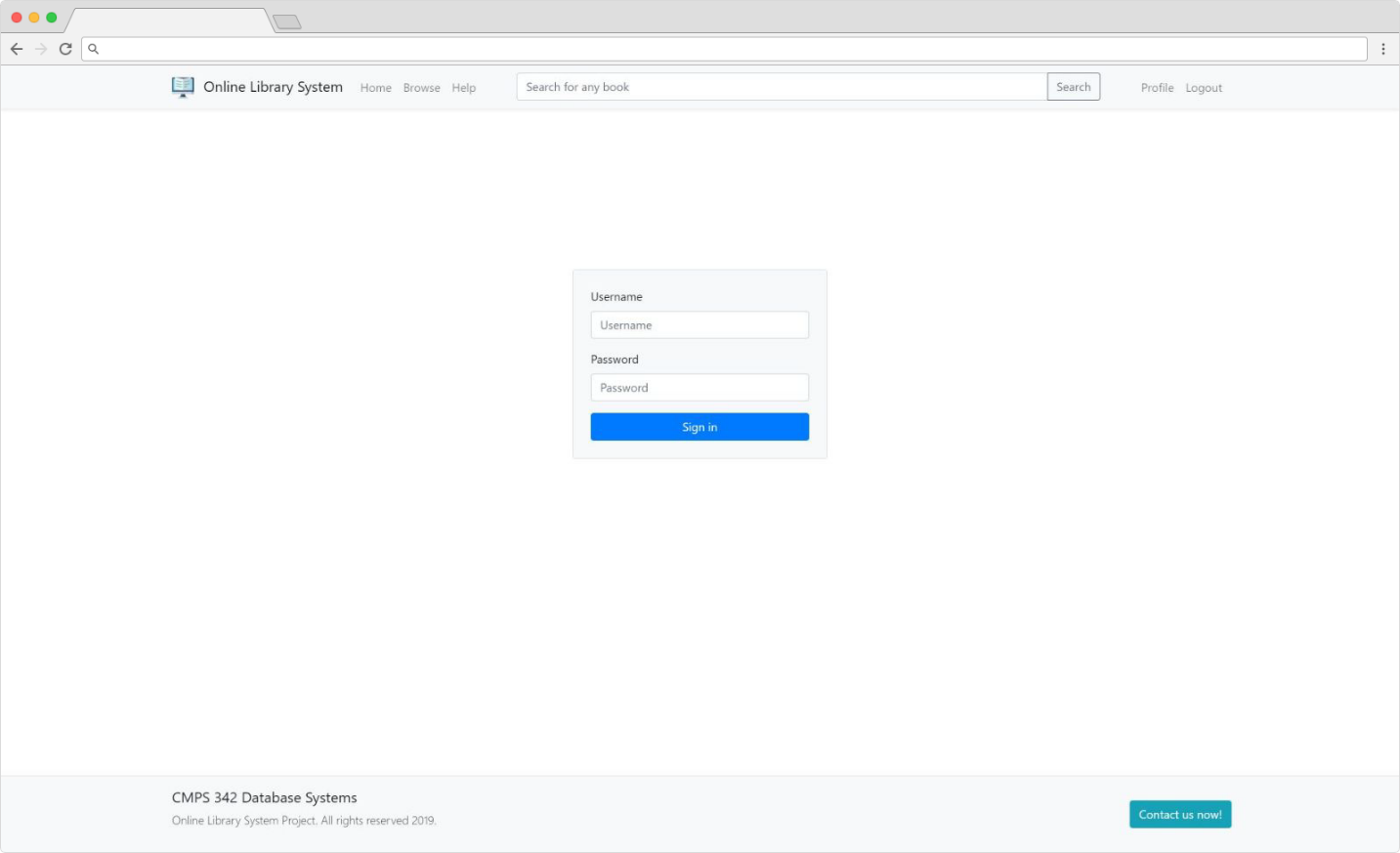
|  |  |
| --- | --- |
| **Page Title** | Online Library | Home Page |
| **Allowed Users** | Guest |
| **Path** | ~/Home/Index |
| **MVC View Path** | ~/Views/Home/Index.cshtml |
| **JavaScript** | ~/Scripts/home.js |



## Login Page

This page presents to the user a simple form that require him to enter his username and password to login.

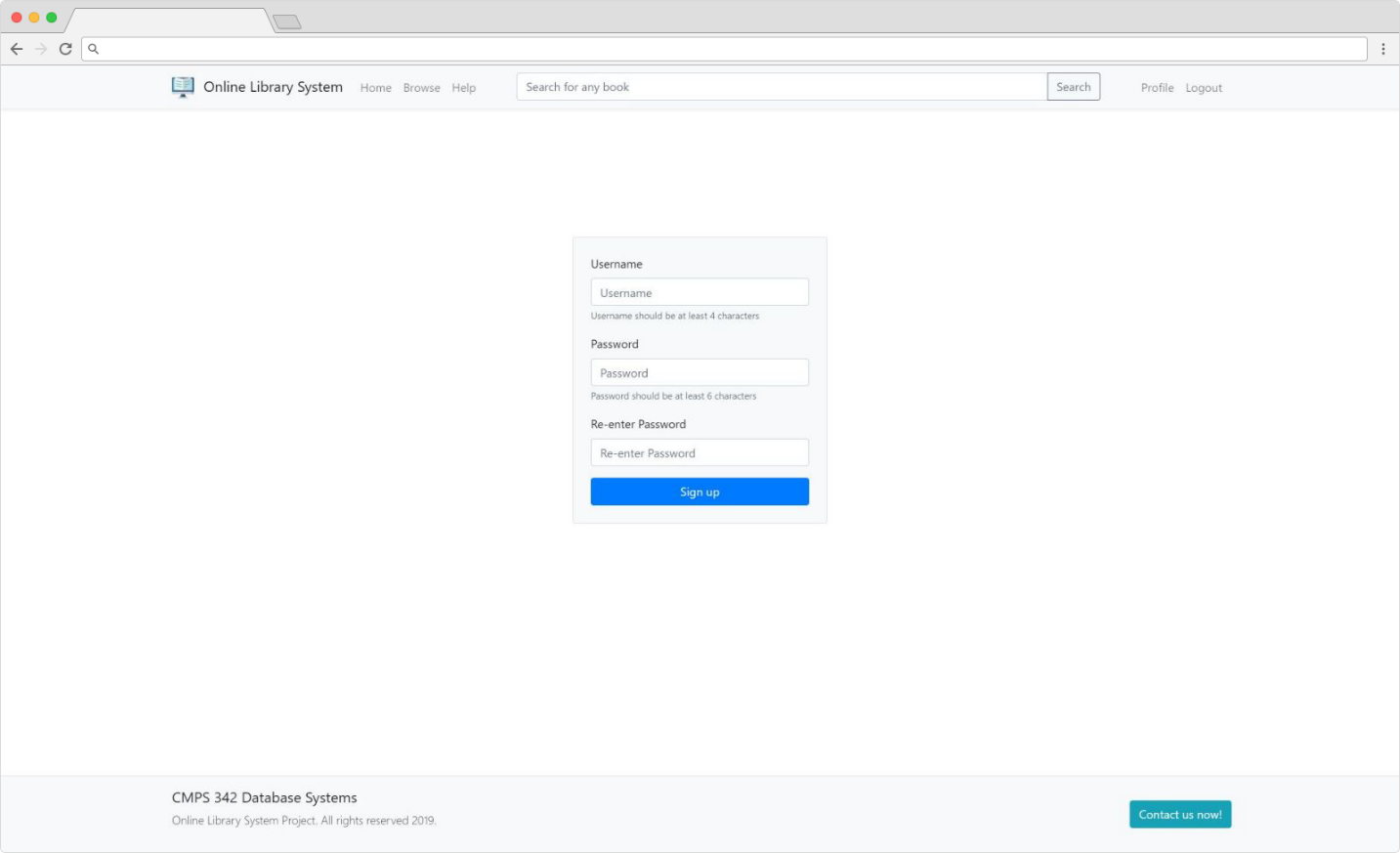
|  |  |
| --- | --- |
| **Page Title** | Online Library | Login |
| **Allowed Users** | Guest |
| **Path** | ~/Account/Login |
| **MVC View Path** | ~/Views/Account/Login.cshtml |
| **JavaScript** | N/A |



## Sign-Up Page

The sign-up page allows students to sign up for an account to be able to rent books.

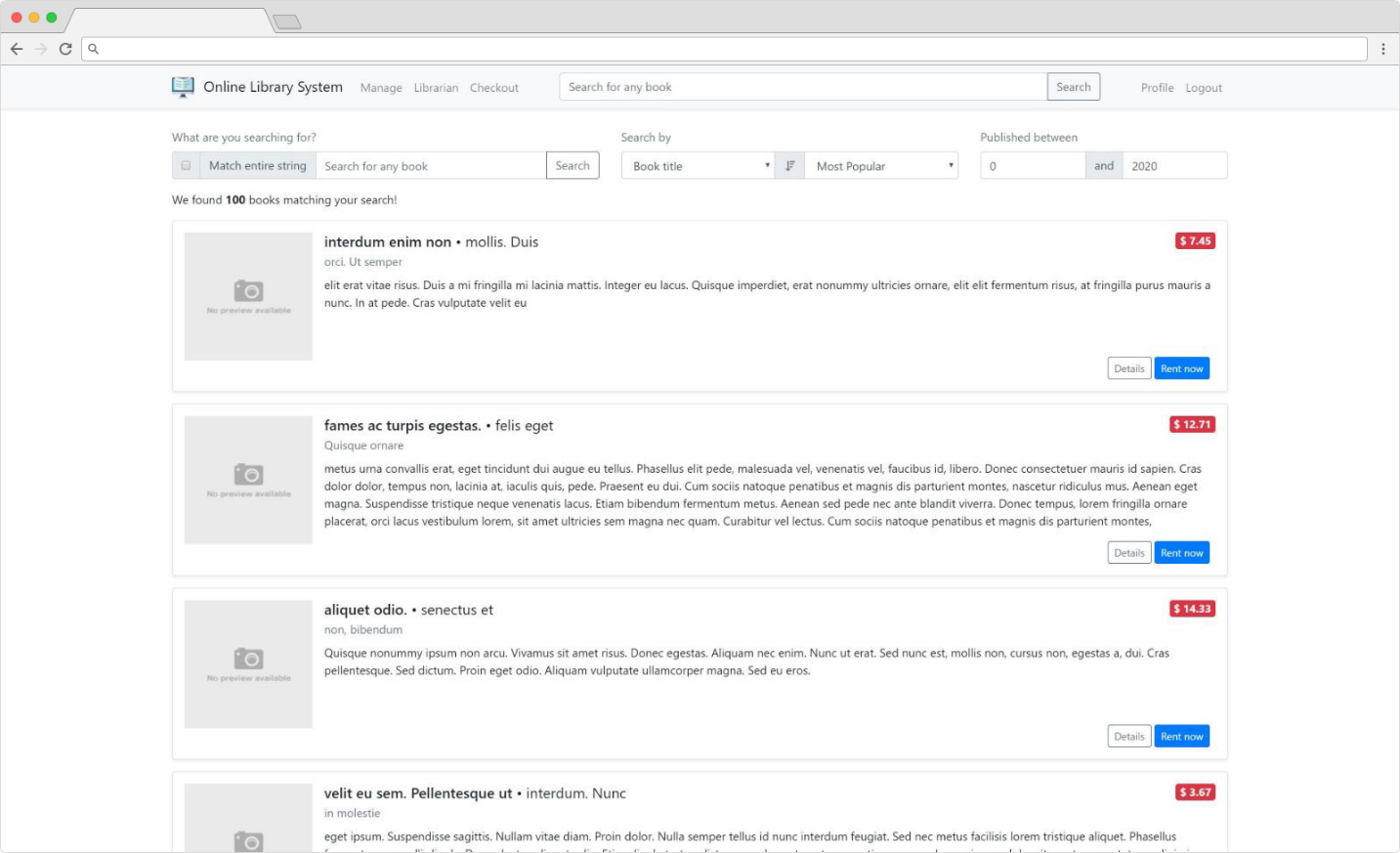
|  |  |
| --- | --- |
| **Page Title** | Online Library | Signup |
| **Allowed Users** | Guest |
| **Path** | ~/Account/Signup |
| **MVC View Path** | ~/Views/Account/Signup.cshtml |
| **JavaScript** | N/A |



## Search Page

This page provides the user with a very detailed but rather easy to use interface to search across a huge database of books.

|  |  |
| --- | --- |
| **Page Title** | Online Library | Search |
| **Allowed Users** | Guest |
| **Path** | ~/Book/Search |
| **MVC View Path** | ~/Views/Book/Search.cshtml |
| **JavaScript** | N/A |

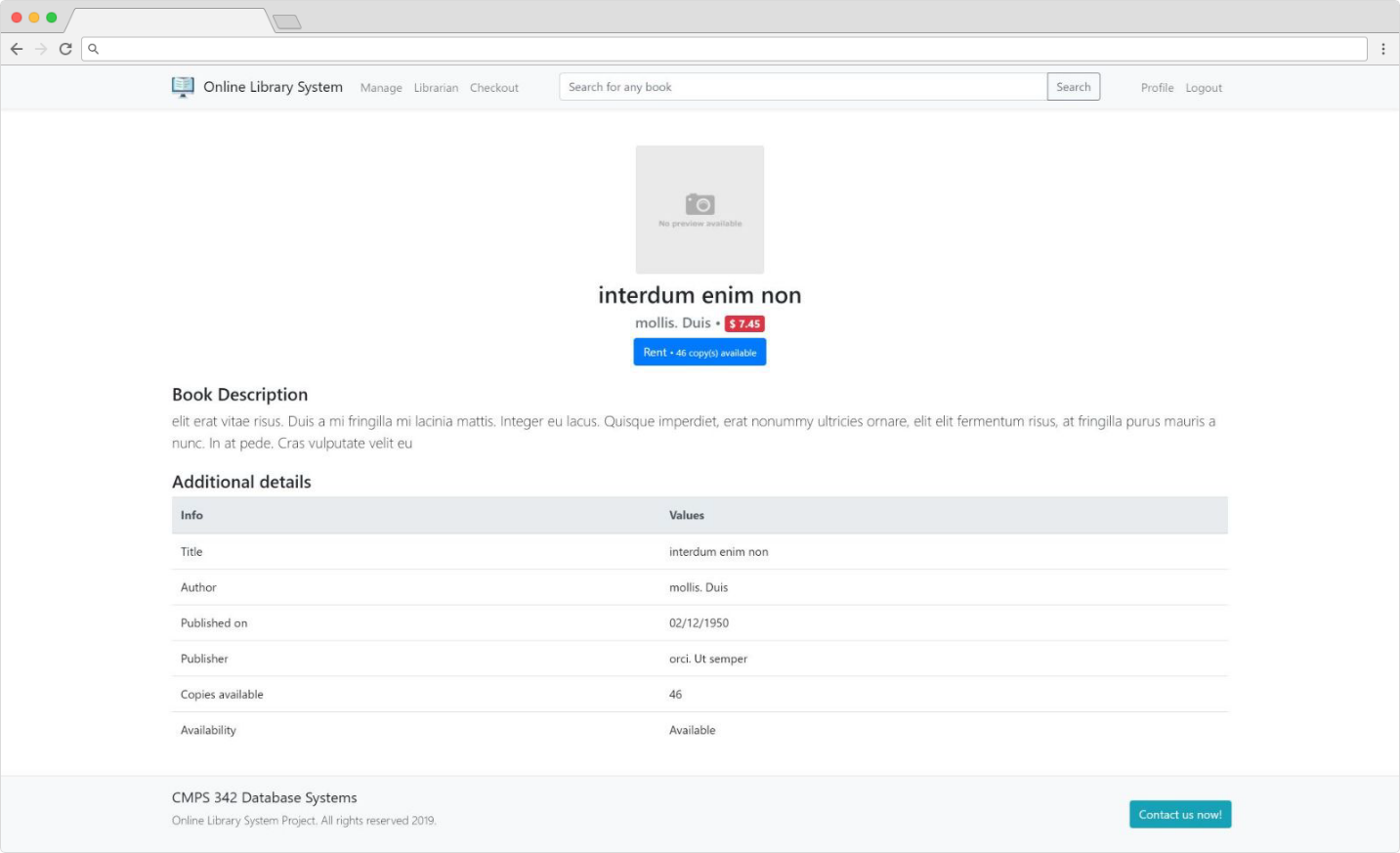


## Book Page

This page shows a detailed description of a book, with popup dialog that shows up when the user clicks on the rent button. The pickup date of a book should be provided so the user can rent it.

|  |  |
| --- | --- |
| **Page Title** | Online Library | ${BookTitle} |
| **Allowed Users** | Guest (only students or higher roles can rent) |
| **Path** | ~/Book/Index |
| **MVC View Path** | ~/Views/Book/Index.cshtml |
| **JavaScript** | N/A |

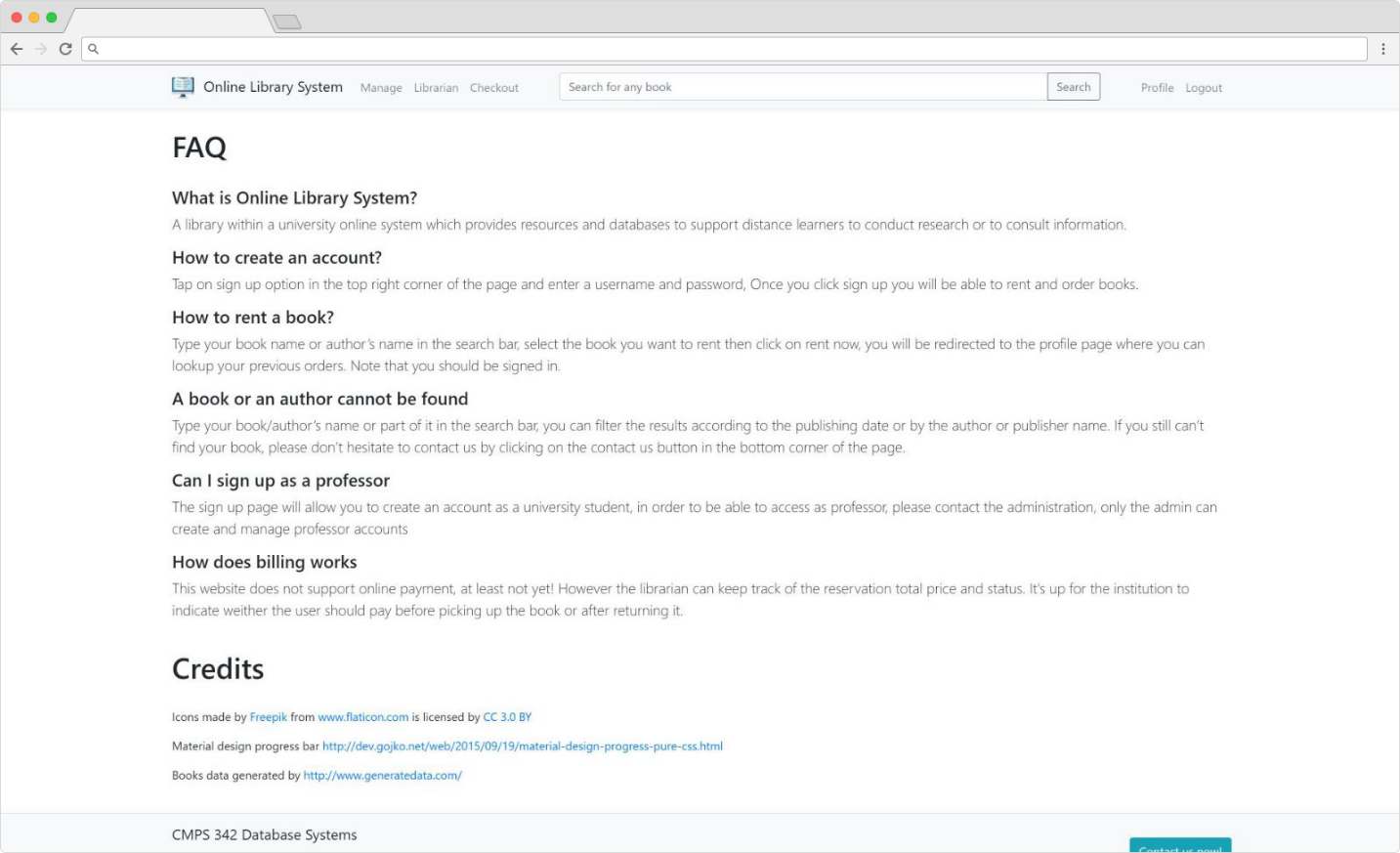
A screenshot of a cell phone

Description generated with very high confidence

## Help Page

This page provides some FAQs (Frequently asked questions) and credits.

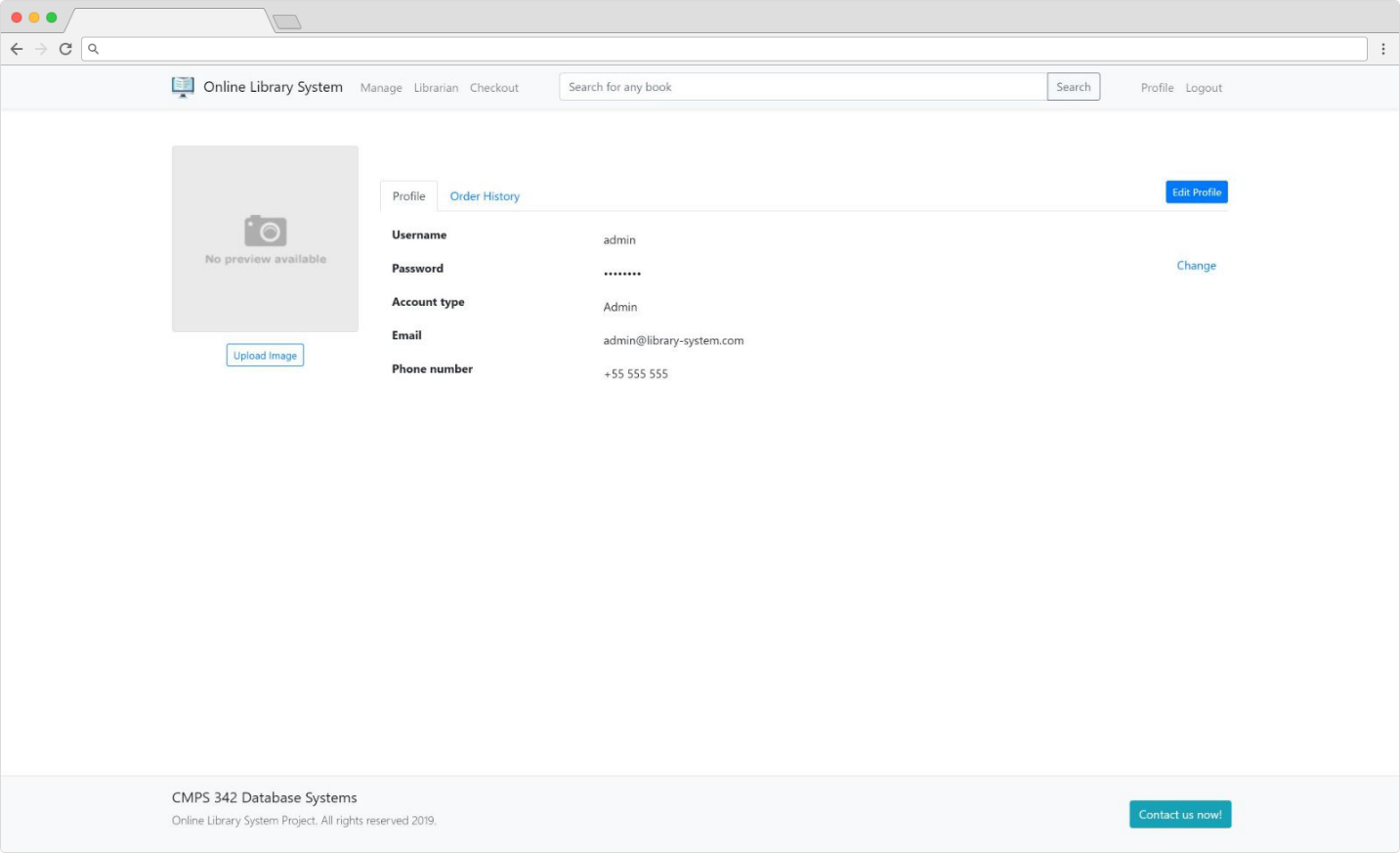
|  |  |
| --- | --- |
| **Page Title** | Online Library | Help |
| **Allowed Users** | Guest |
| **Path** | ~/Book/Help |
| **MVC View Path** | ~/Views/Help/Index.cshtml |
| **JavaScript** | ~/Scripts/help.js |



## Profile Page

A simple page allowing users to check and edit their profile information along with checking previously ordered books.

|  |  |
| --- | --- |
| **Page Title** | Online Library | Profile |
| **Allowed Users** | Students or higher roles |
| **Path** | ~/Account/Profile |
| **MVC View Path** | ~/Views/Account/Profile.cshtml |
| **JavaScript** | ~/Scripts/profile.js |



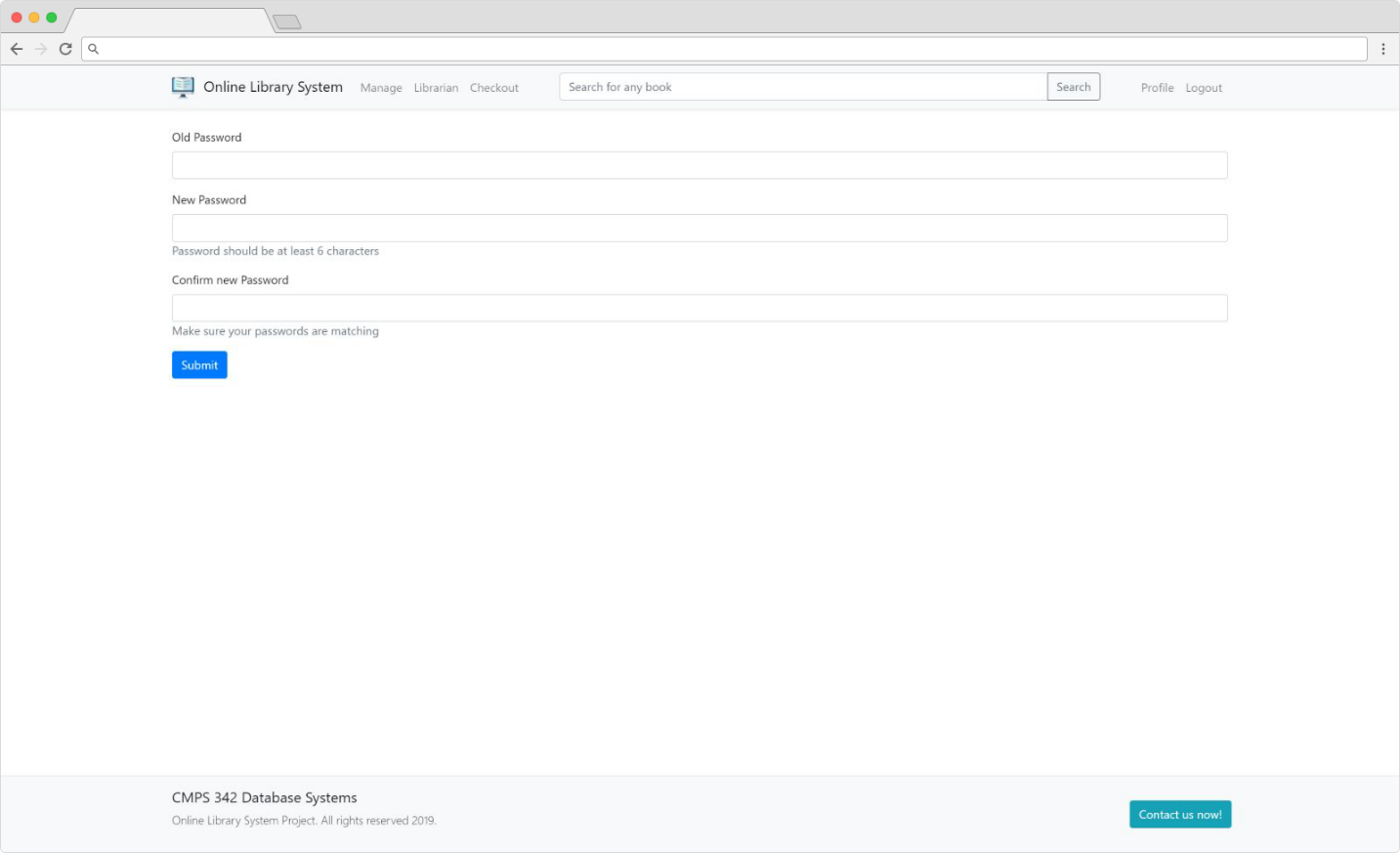
A screenshot of a cell phone

Description generated with very high confidence

## Change Password Page

This page allows users to modify their password securely.

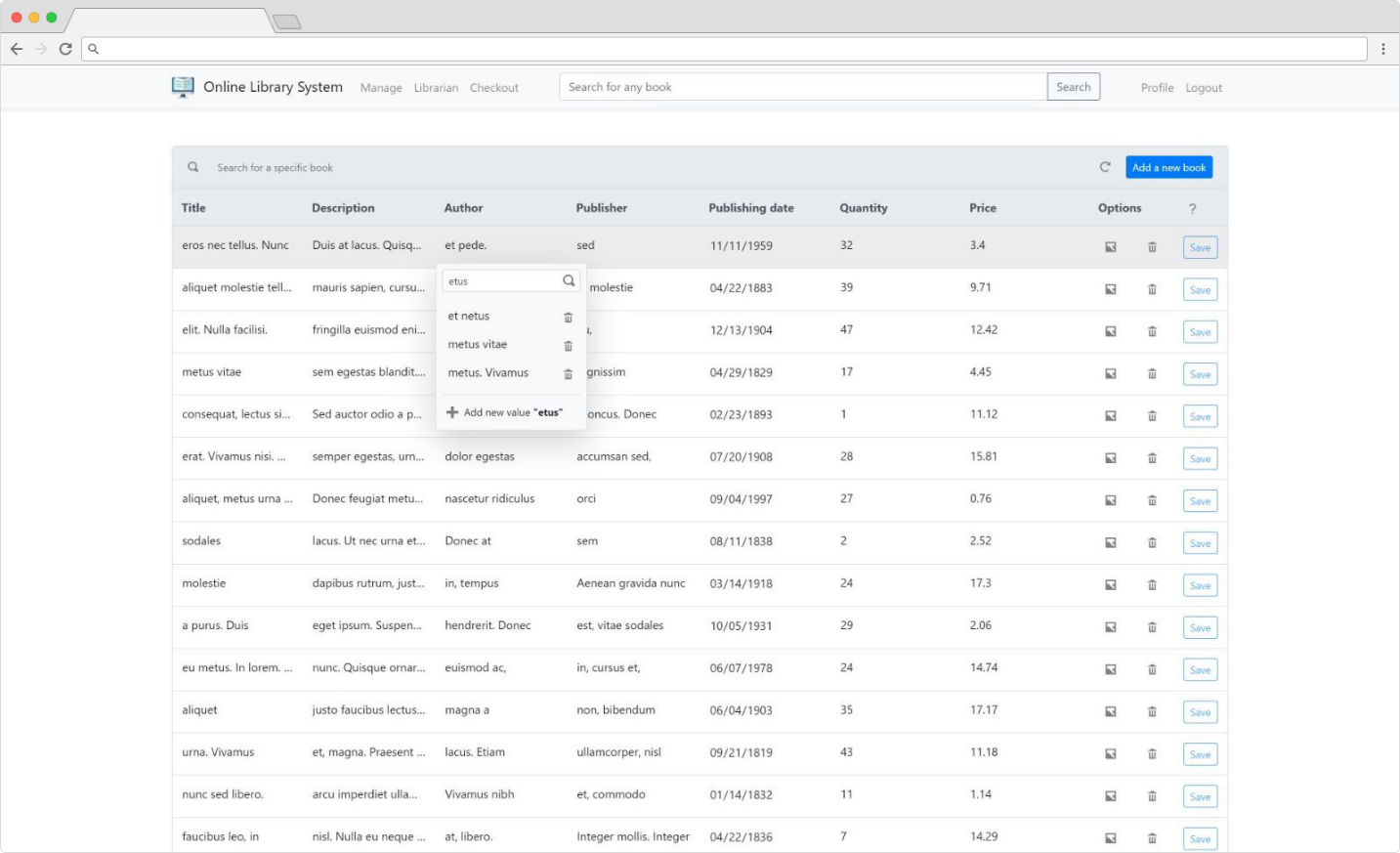
|  |  |
| --- | --- |
| **Page Title** | Online Library | Change Password |
| **Allowed Users** | Guest |
| **Path** | ~/Account/ChangePassword |
| **MVC View Path** | ~/Views/Account/ChangePassword.cshtml |
| **JavaScript** | N/A |



## Librarian Page

A compact interface is provided for the librarian to modify and add books very easily, he can also search by typing a books name. Adding and removing authors and publishers is also a very easy operation.

|  |  |
| --- | --- |
| **Page Title** | Online Library | Librarian |
| **Allowed Users** | Librarians or higher roles |
| **Path** | ~/Librarian/Index |
| **MVC View Path** | ~/Views/Librarian/Index.cshtml |
| **JavaScript** | ~/Scripts/librarian.js |



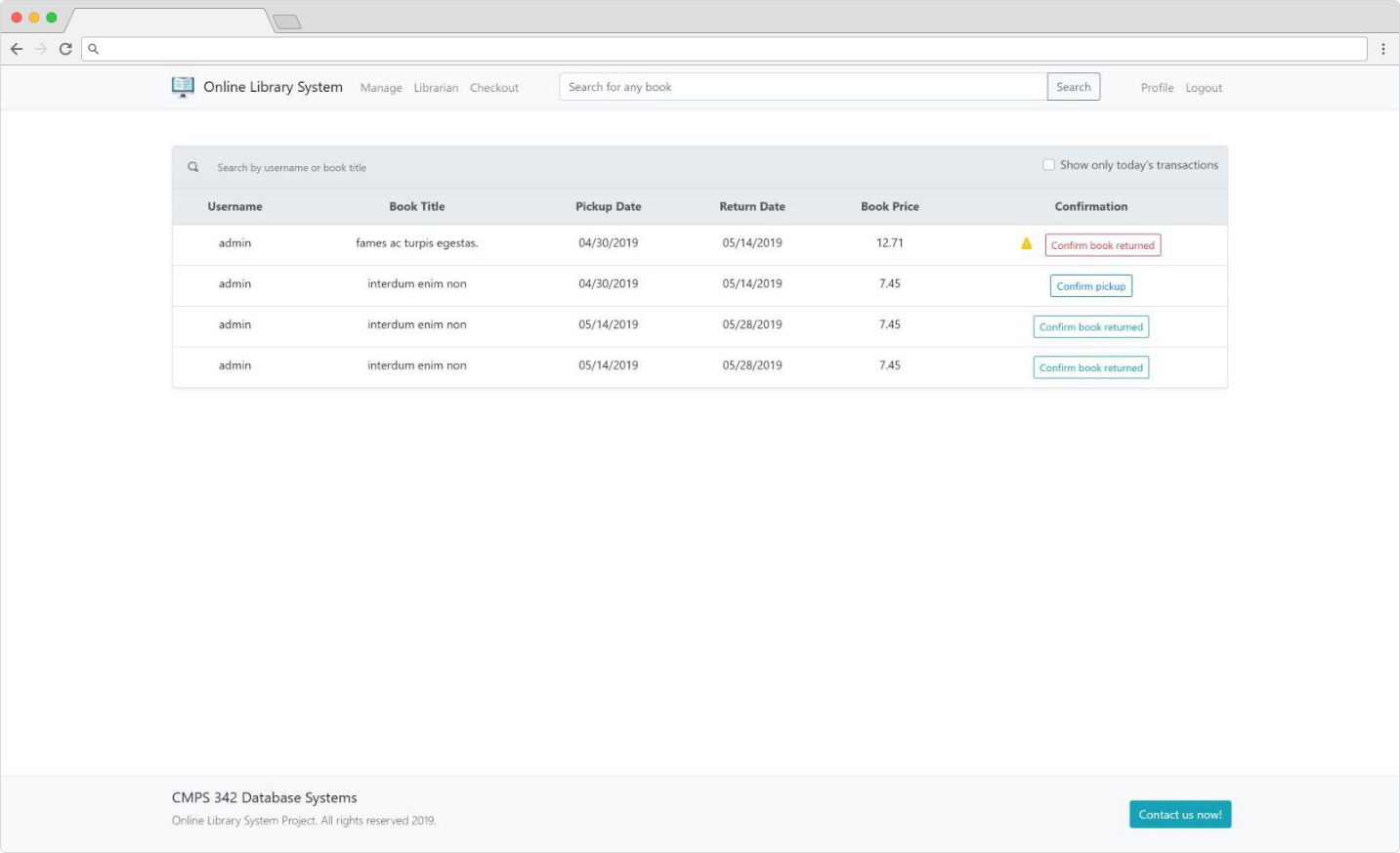
A screenshot of a computer

Description generated with very high confidence

## Checkout Page

This librarian can confirm if a book was picked up or returned in this page, a warning sign will appear next to the late transactions.

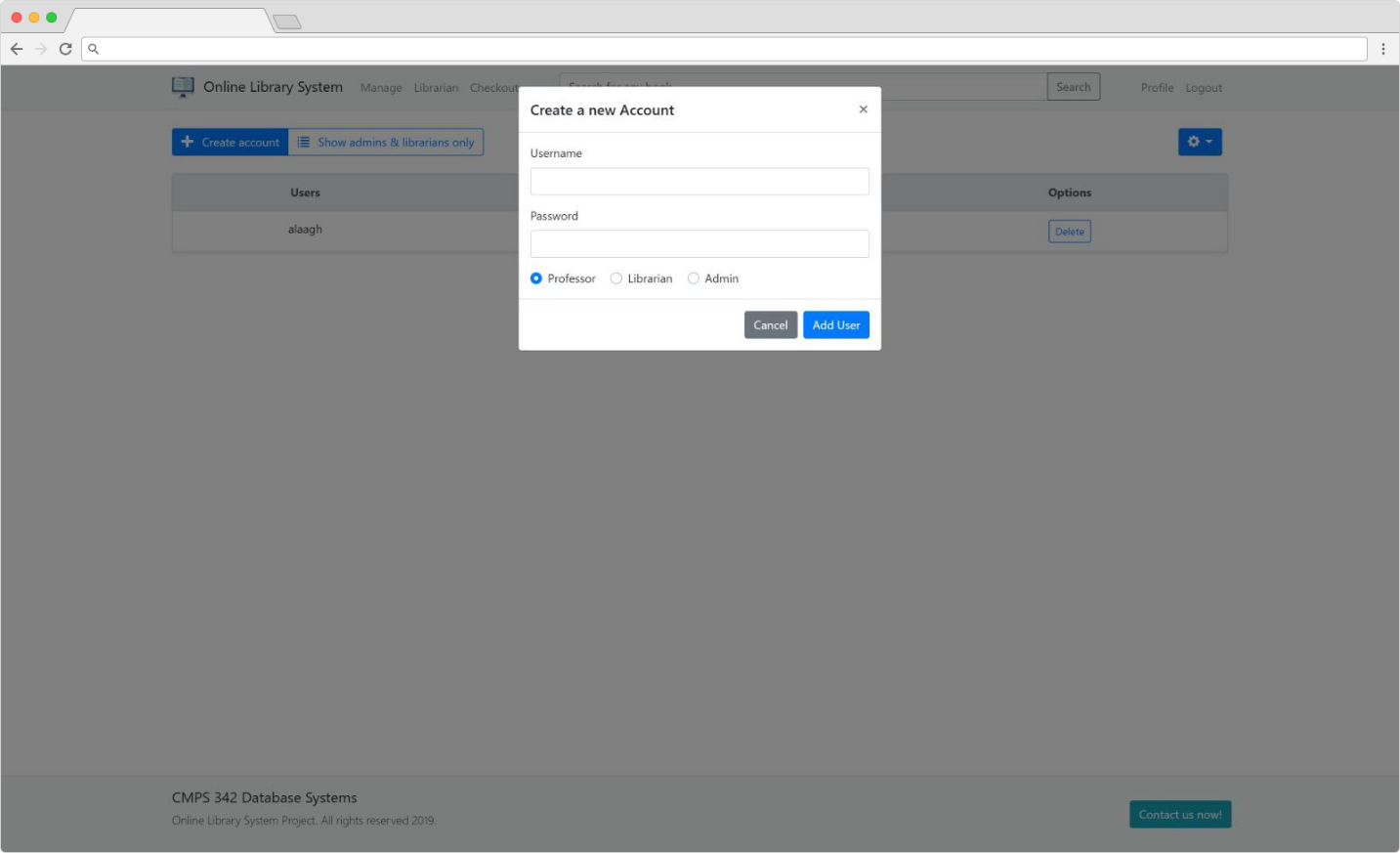
|  |  |
| --- | --- |
| **Page Title** | Online Library | Checkout |
| **Allowed Users** | Librarians or higher roles |
| **Path** | ~/Librarian/Checkout |
| **MVC View Path** | ~/Views/Librarian/Checkout.cshtml |
| **JavaScript** | ~/Scripts/checkout.js |



## Admin Page

Only admins can access this page, this page is used to add new staff (Professors, Librarians or Admins) or delete them (this will keep a copy of the data in the database, but the deleted user will not be able to login anymore).

|  |  |
| --- | --- |
| **Page Title** | Online Library | Manage |
| **Allowed Users** | Admins |
| **Path** | ~/Admin/Manage |
| **MVC View Path** | ~/Views/Admin/Manage.cshtml |
| **JavaScript** | ~/Scripts/checkout.js |



# Brief explanation

## Technologies used

We decided to create this project using ASP.NET MVC, because of the design architecture that allow us to separate the backend from the frontend and make the project easily understood and modified. So now our code is nicely divided into “ApiController” classes and each class is divided into several methods or “actions”. The following are the controllers that can be found in [~/Controllers/](OnlineLibrarySystem/Controllers) folder:

* ApiBookController.cs
  + All actions in this class are related to books.
  + The biggest class since most of our queries are related to books.
* ApiAccountController.cs
  + Actions that are related to users’ information.
* ApiReservationController.cs
  + Contains actions to checkout or check existing reservations.

The queries used inside those files are grouped in the file [ApiQueries.sql](SqlScripts/ApiQueries.sql):

A screenshot of a computer

Description generated with very high confidence

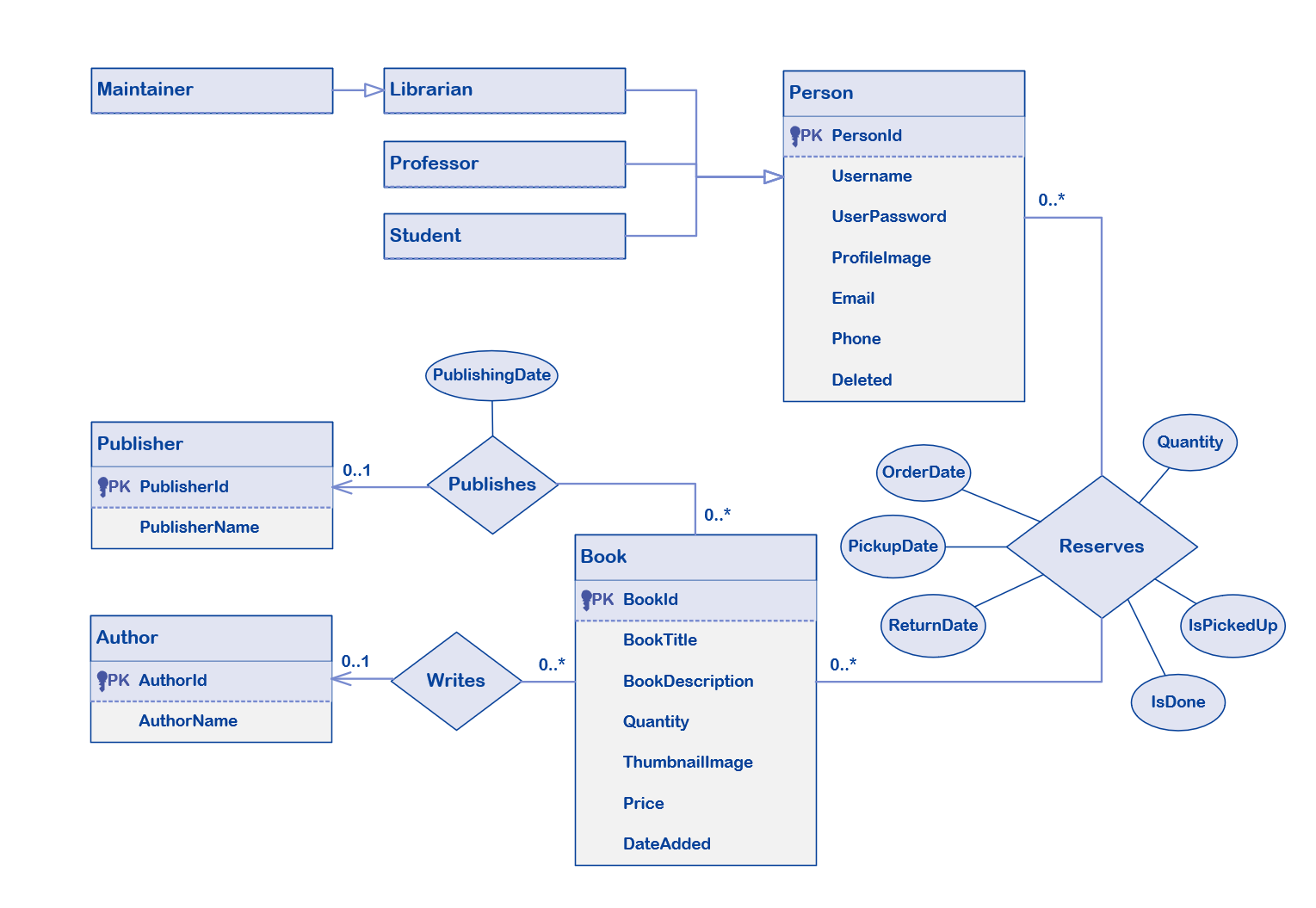
*This file is missing parameters marked as “\_”, running the script will cause errors.*

We also used other technology like angular, allowing us to divide our views into components without redundantly repeating codes.

* **Important:** For more details about the project setup please refer the <readme.md> file.

## Database Design

Here is our **Entity Relations Diagram**:



*We’ve included all diagrams and SQL files inside the visual studio project folder* [*SqlScripts*](SqlScripts)*.*

Our SQL files are split into a [DDL](SqlScripts/DDL.sql) and [DML](SqlScripts/DML.sql) file to keep everything clean. And there’s also an [MDF](SqlScripts/diagrams/OnlineLibrarySystem.mdf) file that can be copied and attached to SQL Server.

We made some design tweaks considering that in real world applications, requirements can change very often. That’s why some tables are made of only one attribute, but only because extending that table and adding more attributes to it in the future is a must (e.g. we might need to store a specific information for professors that is not related to students so we might need to add that later that’s why we kept the professor table separated, or maybe a maintainer can have a creator id which gives him specific permissions or restriction over other admins) but we kept things nice and easy for now!

## Additional Links

* Hosted project on **somee.com**: <http://www.onlinelibrarysystem.somee.com>
* Our **GitHub** repository: <https://github.com/omneimneh/online-library-system>
* Screenshots are located in the <screenshots> folder