# *Library Application*

## 1. Introduction

A library application was created using a modern design approach. The database system uses PostgreSQL, which claims to be the world's most advanced open source database. MySQL could have been used instead, however Postgres has some more features to offer such as special types and schema namespaces. The application interface is Apache Web Server. Using a micro framework called Flask, I implemented the application in Python and deployed it to a linux server through Digital Ocean.

The application has two sets of users. The first group are the administrators and the second are readers. Since both share some common attributes I decided to extend them from another common group in the database called actors. Admins login with an email address and a password. Readers may do the same but also have the option to use their card number instead of their email. Once inside the admin may view information regarding books, branches, readers, inventory, and various statistics. They have the authority to add a reader, and add more book copies to a branch.

Readers can search for books, but in this application they don't see if a book is available at a particular branch. After they search for book matching a particular title or author, they can either checkout that book or they can reserve it. The system makes sure that the reader does not go over a limit of 10 books but the reader will not see that process. The reader should be able to return their books also, however I was not able to implement that on time. Finally they can see the status of their account as a list of checkouts and reservations.

## 2. Database Design

*Legend:*

| Primary Key | Foreign Key | Composite Key | Both Primary / Foreign |
|---|---|---|---|
| **Strong Entity [S]** | **Weak Entity [W]** | **Relation [R]** | |

***Actor[S]***

| | |
|---|---|
| ID | integer |
| name | string |
| email | string |
| phone | integer |
| address | string |

***Admin[W]***

| | |
|---|---|
| ActorID | integer |

***Reader[W]***

| | |
|---|---|
| ActorID | integer |
| card | integer |

***Author[S]***

| | |
|---|---|
| ID | integer |
| name | string |

***Publisher[S]***

| | |
|---|---|
| ID | integer |
| name | string |
| adress | string |

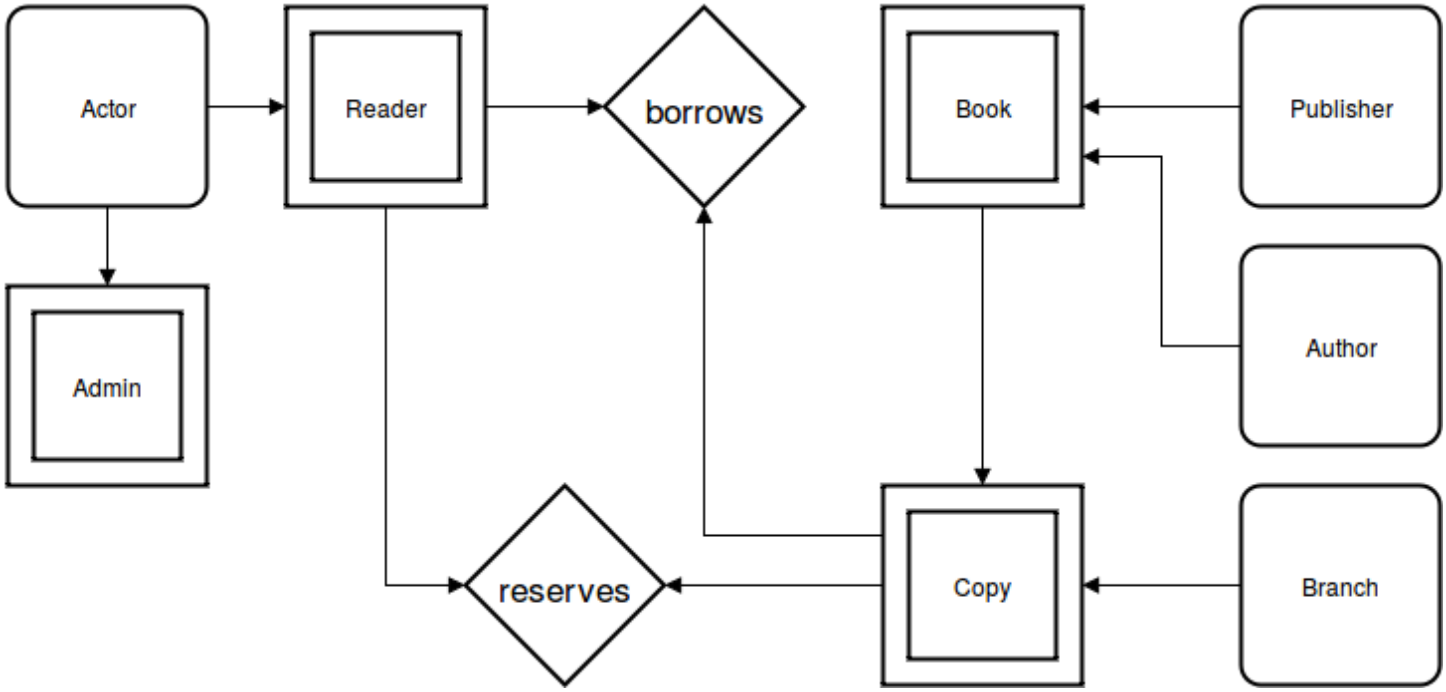
| ***Book[W]*** | ***Branch[S]*** | | ***Copy[R]*** |
|---|---|---|---|
| ISBN integer<br>pubdate date<br>title string<br>AuthorID integer<br>PublishID integer | ID integer<br>name string<br>address string | | ID integer<br>ISBN integer<br>BranchID integer<br>code integer<br>lock bool |

| ***Borrow[R]*** | | ***Reserve[R]*** | |
|---|---|---|---|
| CopyID integer<br>ReaderID integer<br>pickup date<br>return date<br>payment money | | CopyID integer<br>ReaderID integer | |

*Relations:*

| | | | | |
|---|---|---|---|---|
| Admin | (1, 1) | *is* | (1, 1) | Actor |
| Admin | (1, 1) | *is* | (1, 1) | Actor |
| Author | (0, N) | *writes* | (0, 1) | Book |
| Publisher | (0, N) | *publishes* | (0, 1) | Book |
| Branch | (0, N) | *shelves* | (0, M) | Book |
| Reader | (0, 1) | *borrows* | (0, 10) | Book Copy |
| Reader | (0, 1) | *reserves* | (0, 10) | Book Copy |

*Initial Entity Relation Diagram:*




### 3. SQL Statements

The code for database creation is provided as two files. ***Rebuild.sql*** destroys any schemas within the database called csci760 and then recreates them and their corresponding tables, views, and functions. ***Inflate.sql*** should be executed after rebuilding is complete. This script creates some test users and fills the library systems with some branches and books. Each branch gets 10 copies of every book by default. The functional statements that query the database are all generated on the fly in a python file called ***base.py***.

### 4. Installation

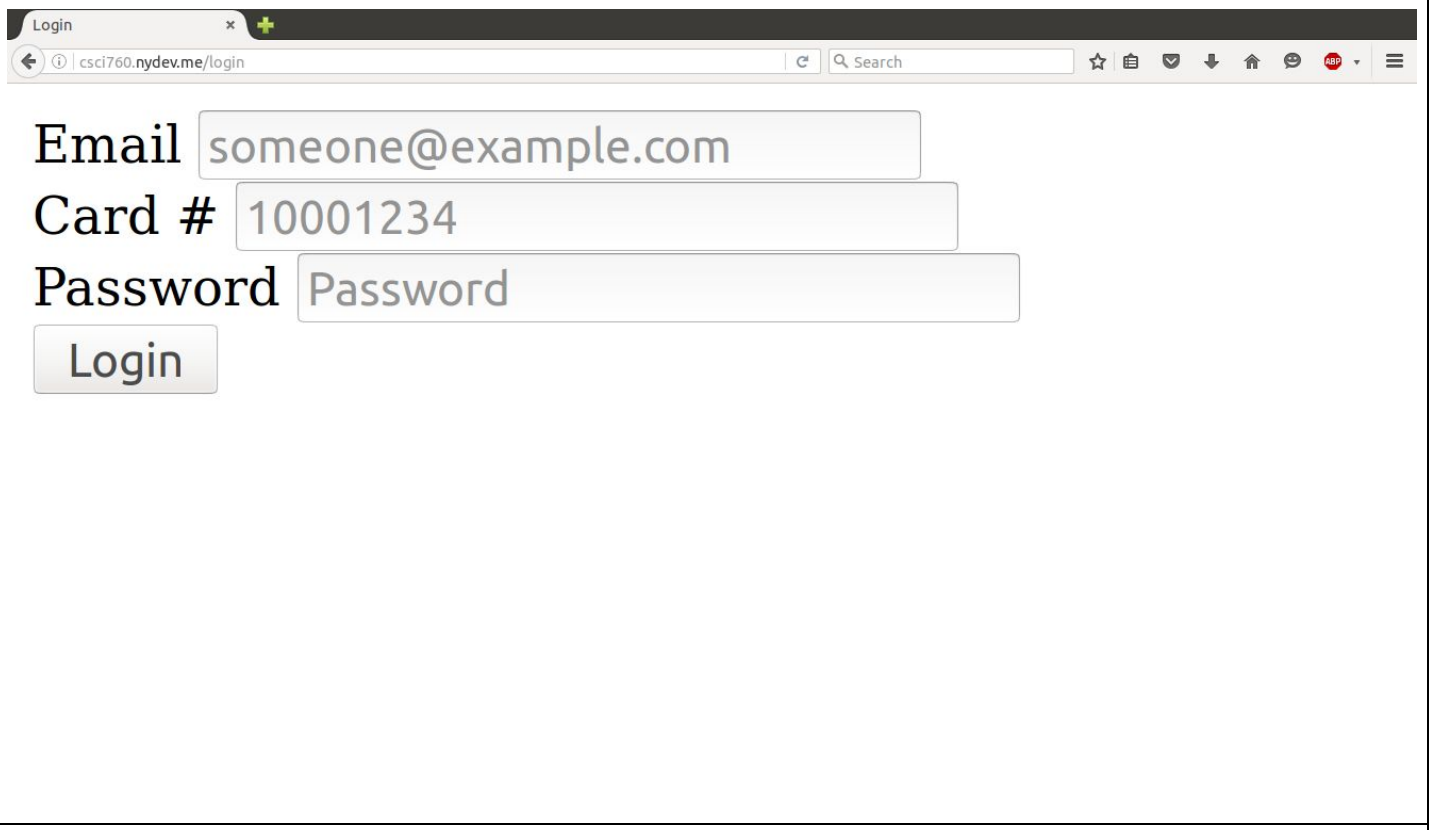
*Requirements:*

1. Debian Linux Operating System
2. PostgreSQL
3. Python3
4. Flask
5. Apache
6. WSGI

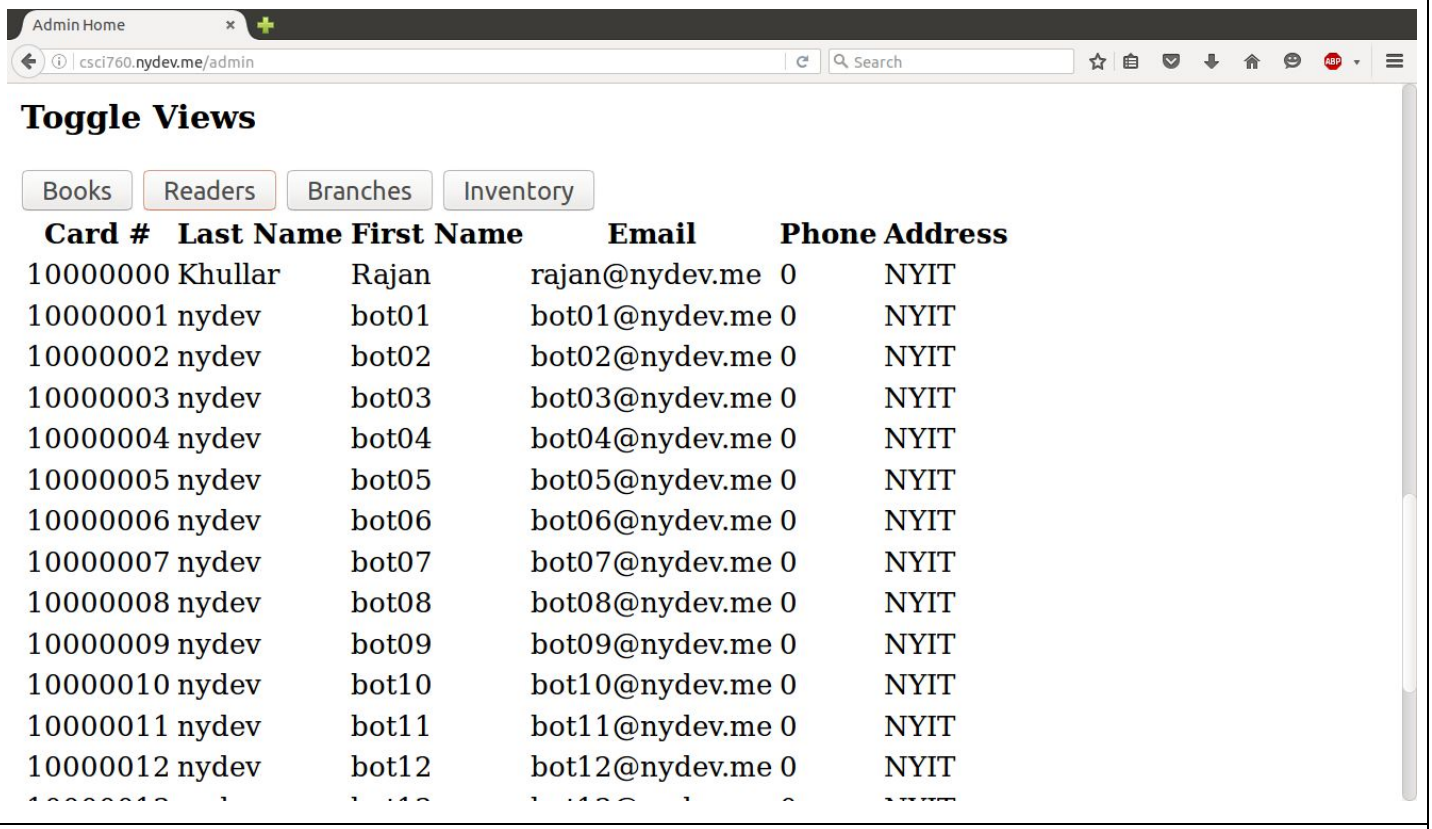
*Instructions:*

1. As the postgres user create your database owned by an appropriate user.
2. Execute the rebuild and inflate scripts.
3. Extract zip project to an appropriate location under /srv.
4. Create an apache virtual host file to serve the web-application.

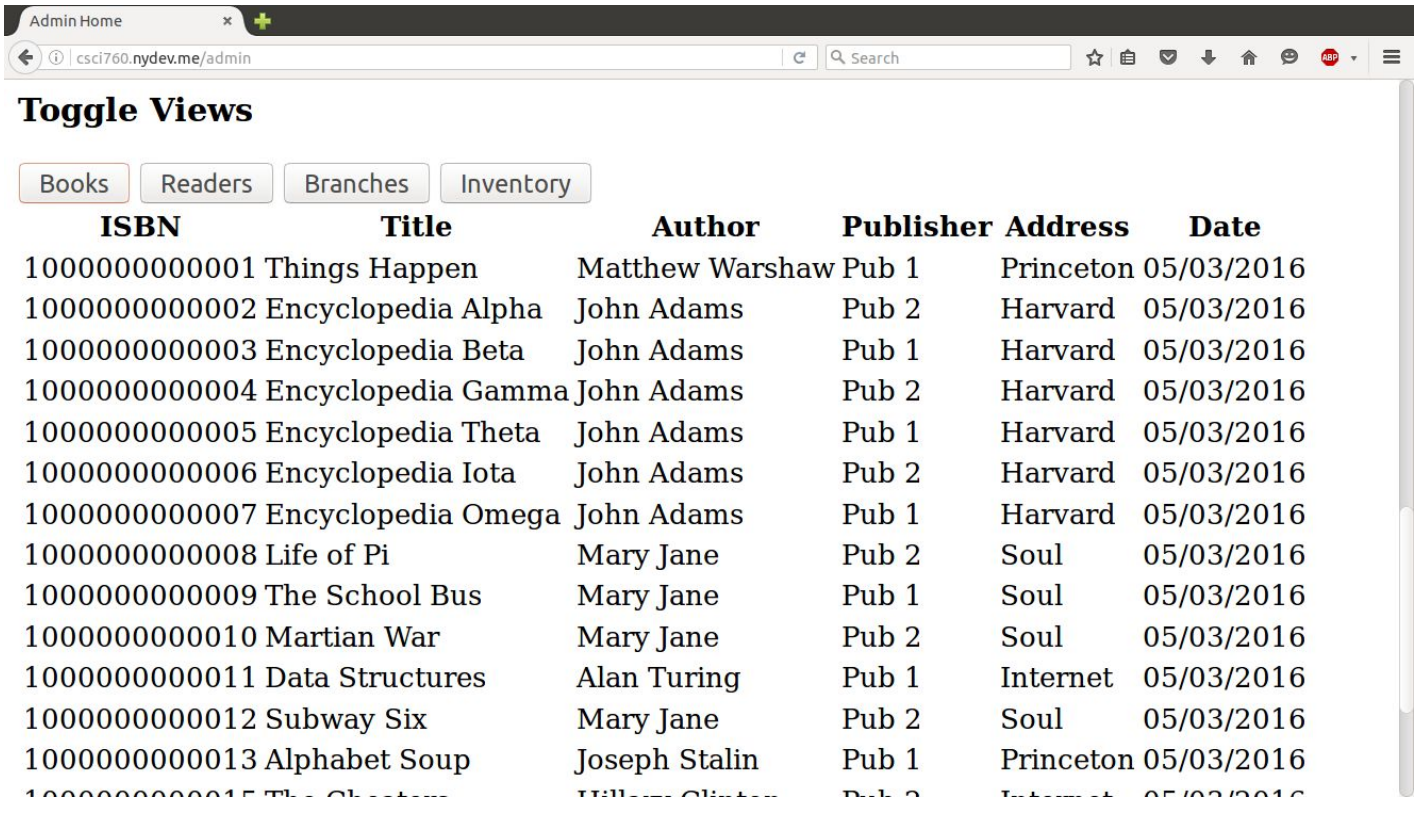
**5. Appendix**




*Login Page*

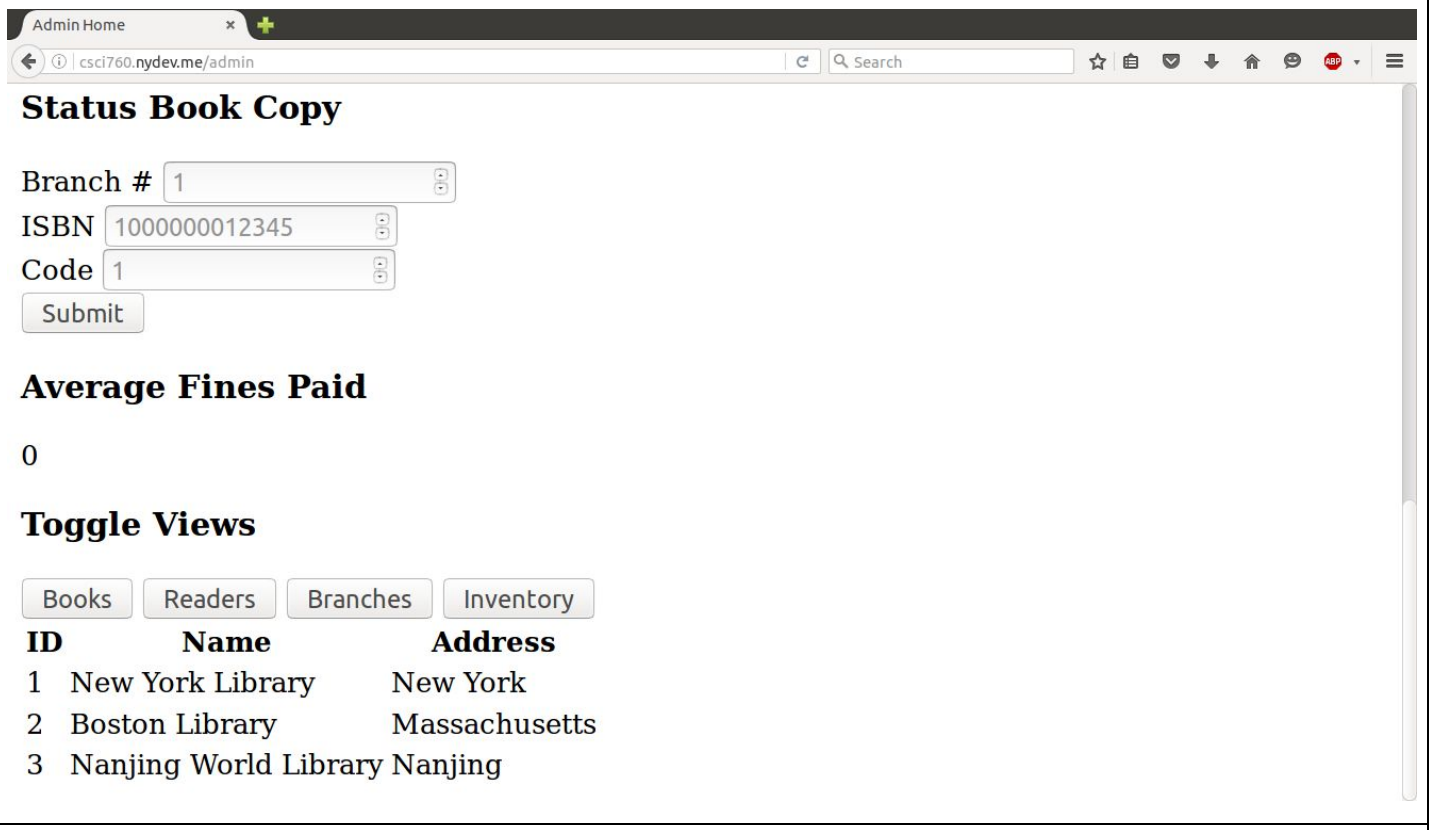



*Admin View Readers*

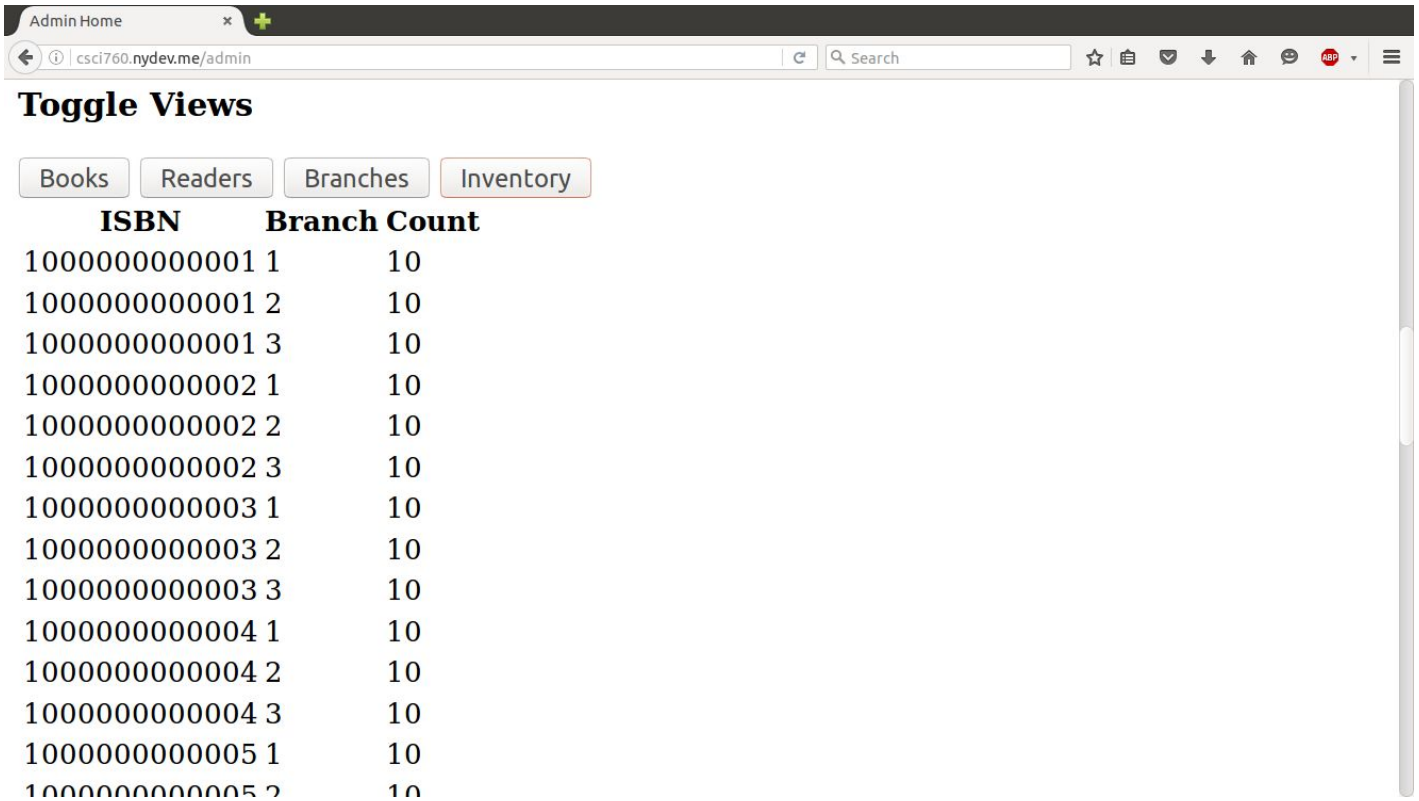



| ISBN | Title | Author | Publisher | Address | Date |
|---|---|---|---|---|---|
| 1000000000001 | Things Happen | Matthew Warshaw | Pub 1 | Princeton | 05/03/2016 |
| 1000000000002 | Encyclopedia Alpha | John Adams | Pub 2 | Harvard | 05/03/2016 |
| 1000000000003 | Encyclopedia Beta | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000004 | Encyclopedia Gamma | John Adams | Pub 2 | Harvard | 05/03/2016 |
| 1000000000005 | Encyclopedia Theta | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000006 | Encyclopedia Iota | John Adams | Pub 2 | Harvard | 05/03/2016 |
| 1000000000007 | Encyclopedia Omega | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000008 | Life of Pi | Mary Jane | Pub 2 | Soul | 05/03/2016 |
| 1000000000009 | The School Bus | Mary Jane | Pub 1 | Soul | 05/03/2016 |
| 1000000000010 | Martian War | Mary Jane | Pub 2 | Soul | 05/03/2016 |
| 1000000000011 | Data Structures | Alan Turing | Pub 1 | Internet | 05/03/2016 |
| 1000000000012 | Subway Six | Mary Jane | Pub 2 | Soul | 05/03/2016 |
| 1000000000013 | Alphabet Soup | Joseph Stalin | Pub 1 | Princeton | 05/03/2016 |

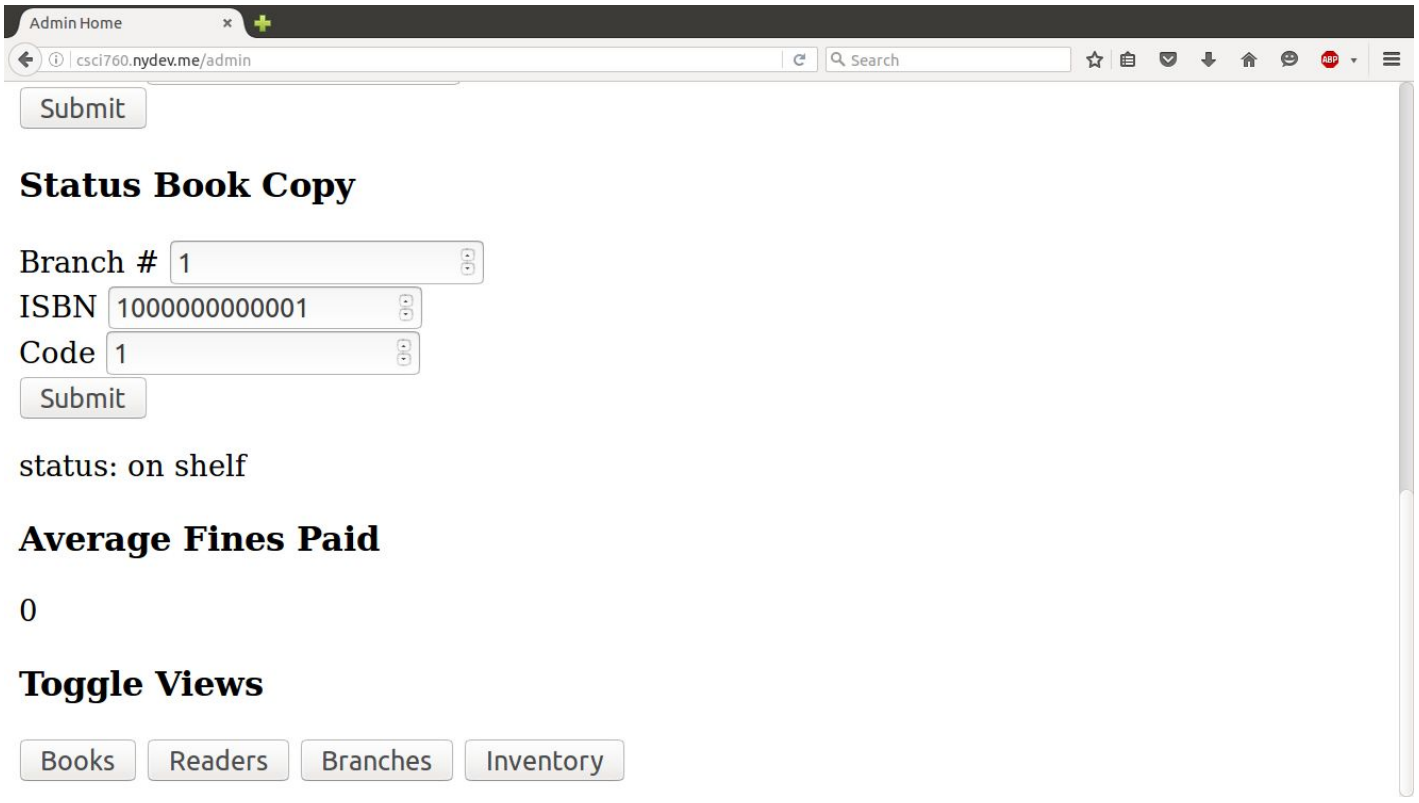
*Admin View Books*



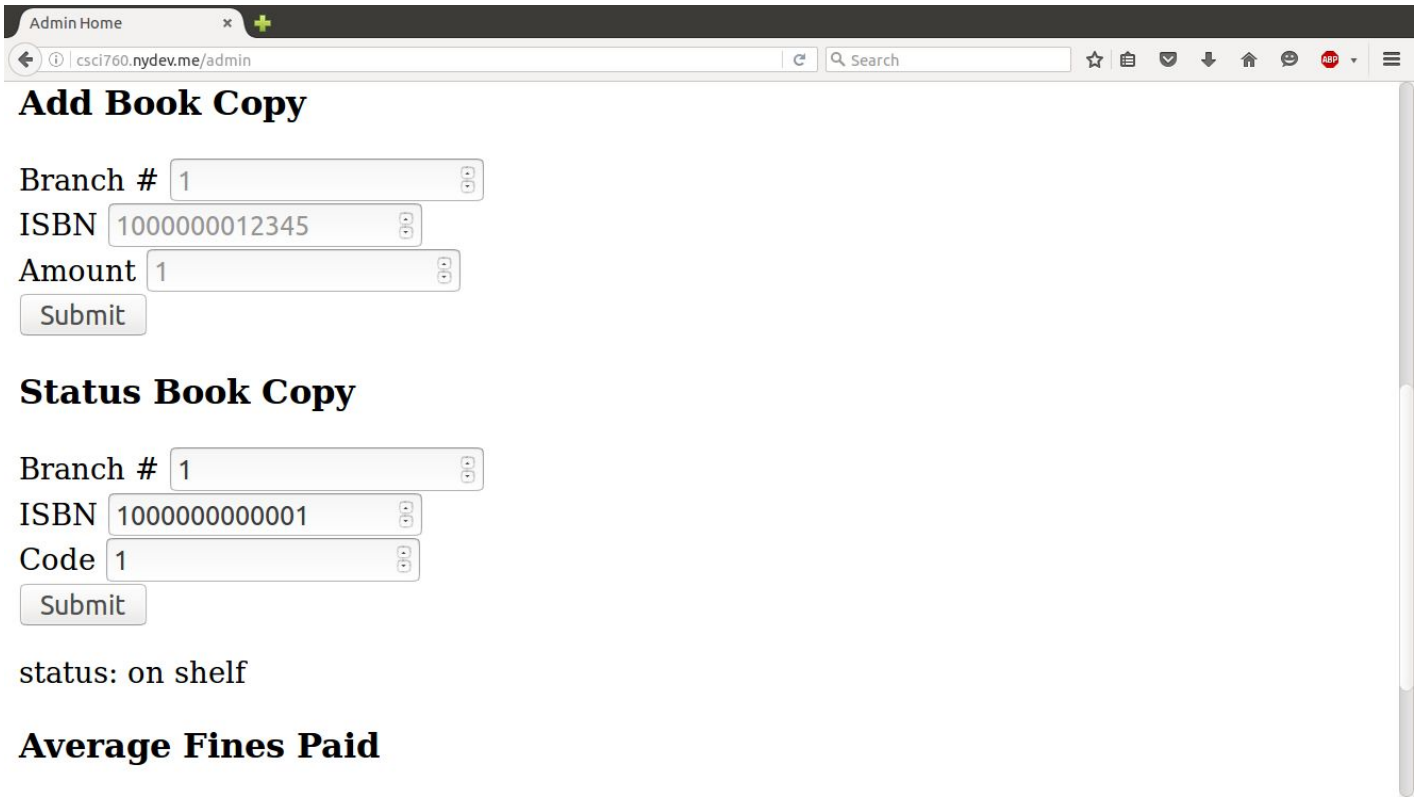

*Admin View Branches*



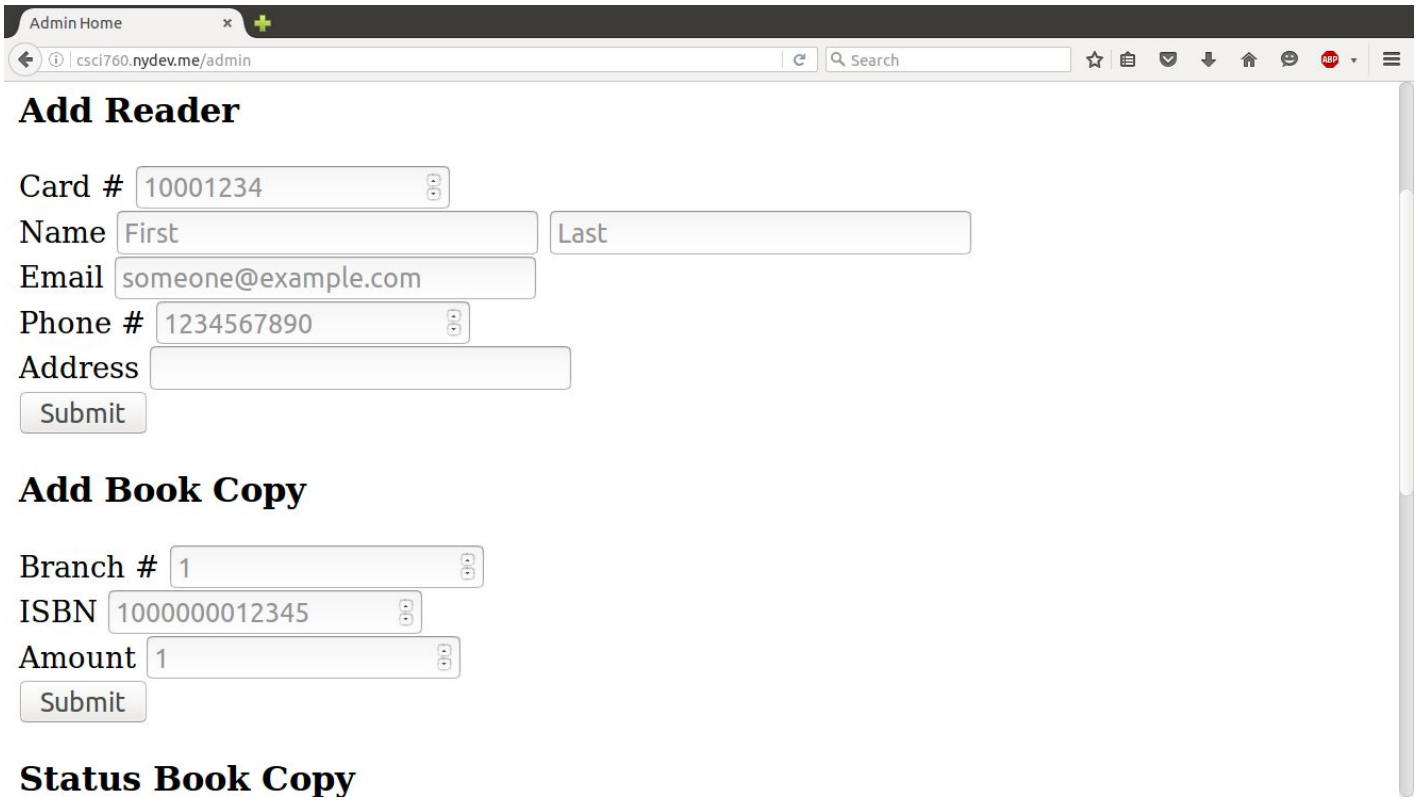

*Admin View Inventory*



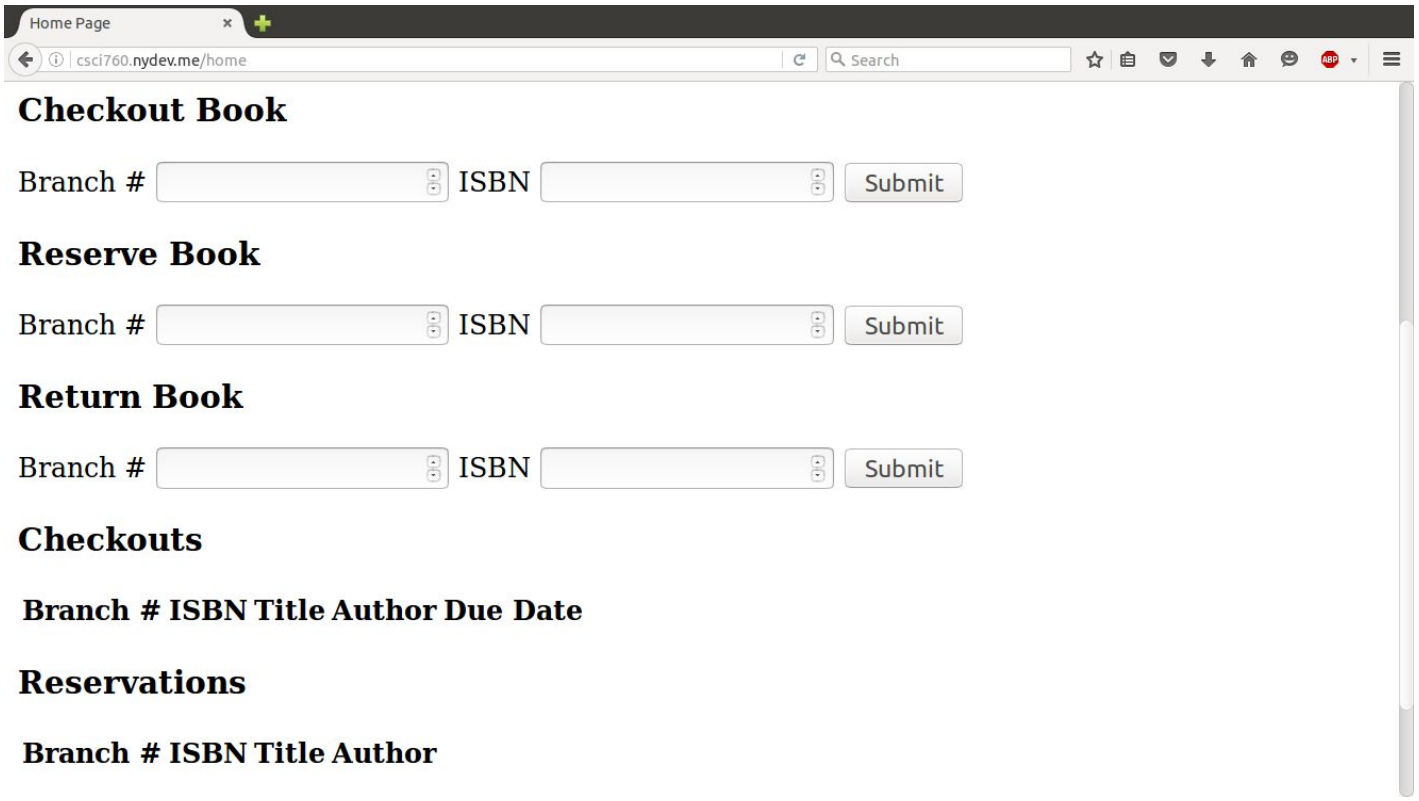

*Admin Book Copy Status*



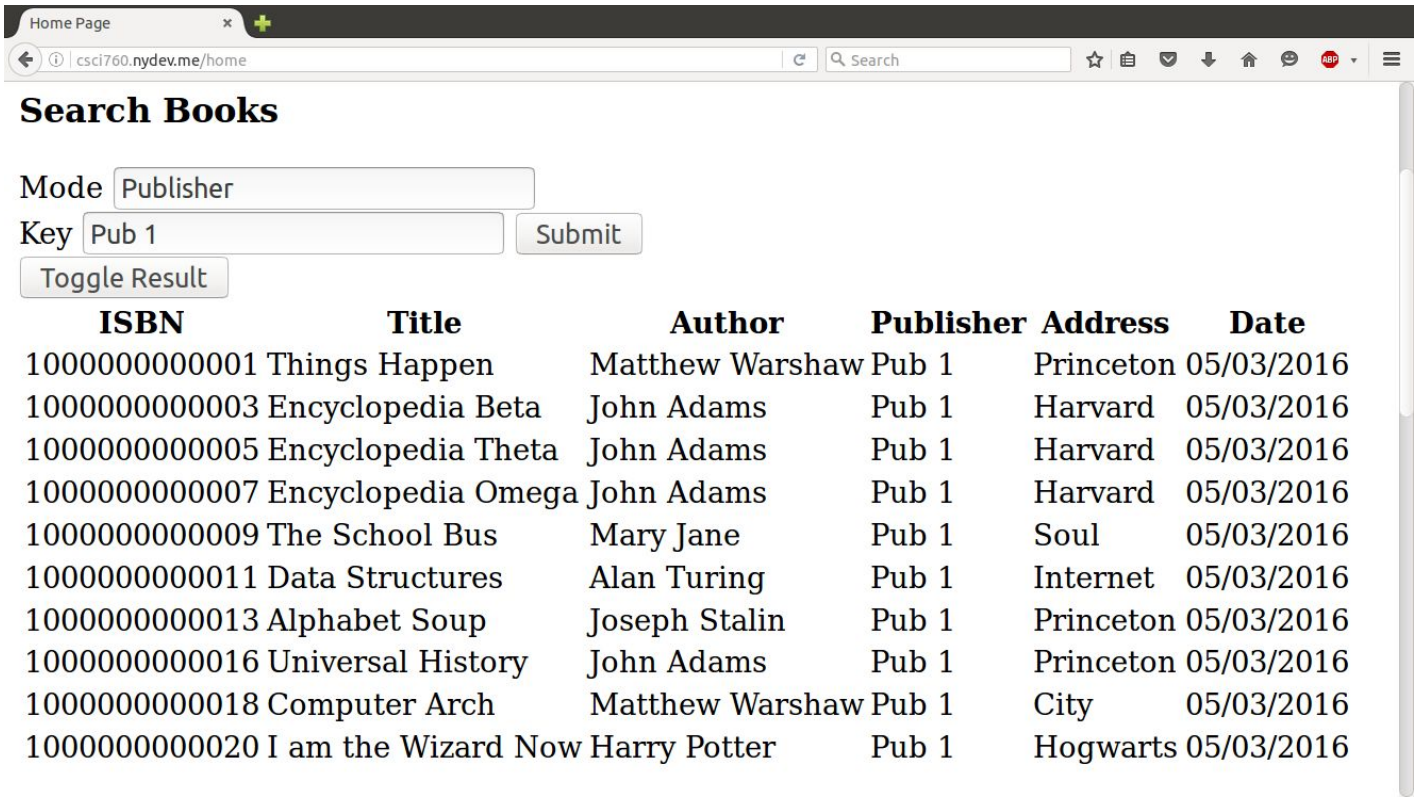

*Admin Add Book Copies*




*Admin Add New Reader*

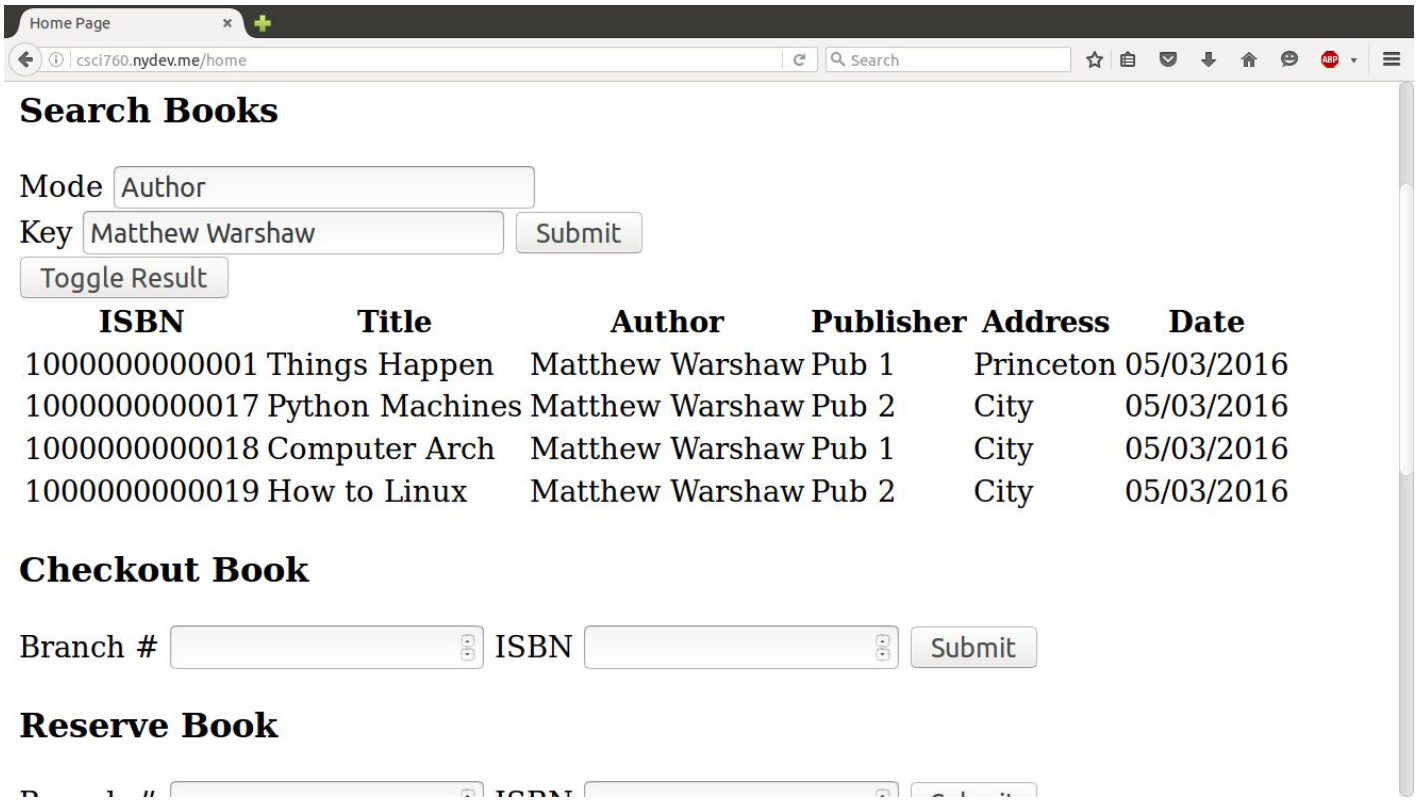



*Reader General Functions*

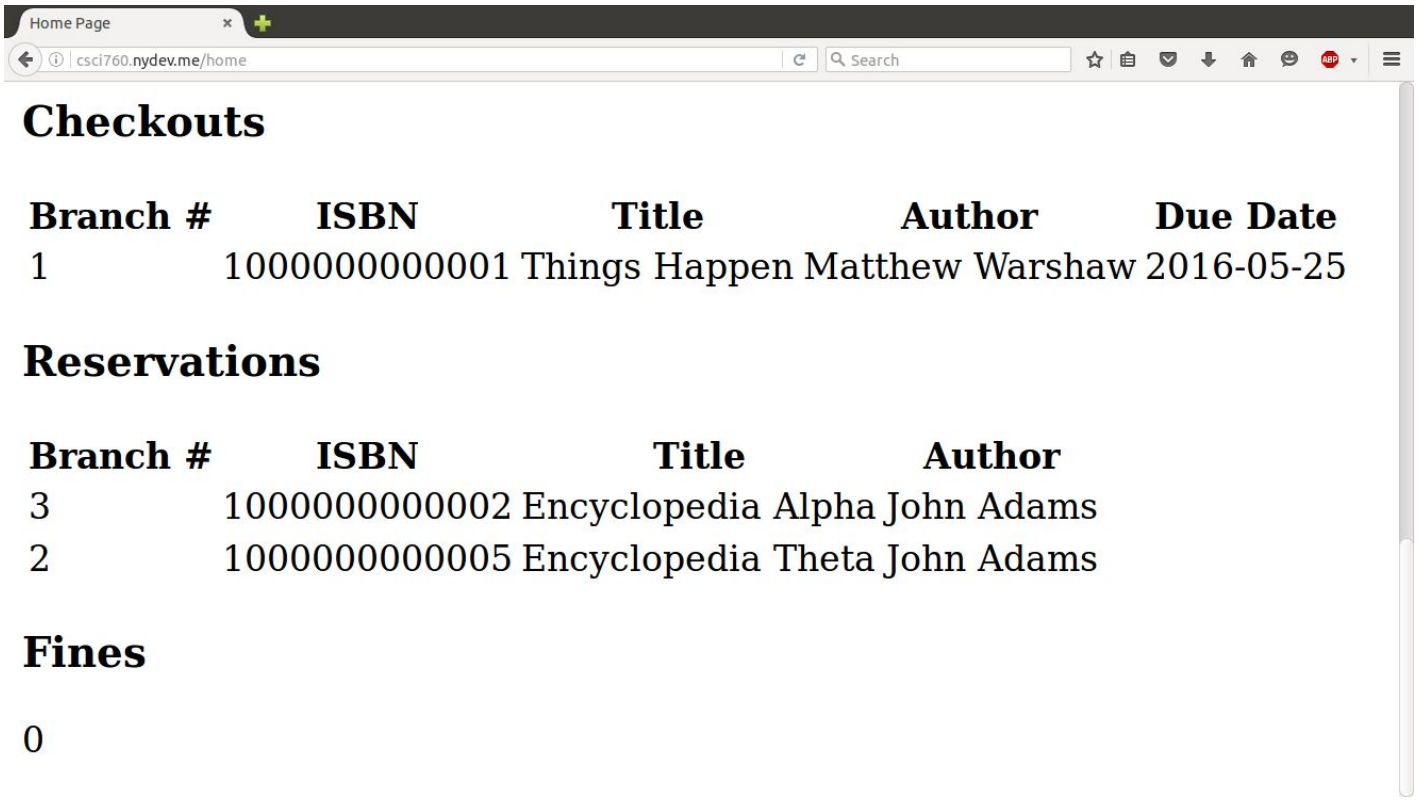



| ISBN | Title | Author | Publisher | Address | Date |
|---|---|---|---|---|---|
| 1000000000001 | Things Happen | Matthew Warshaw | Pub 1 | Princeton | 05/03/2016 |
| 1000000000003 | Encyclopedia Beta | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000005 | Encyclopedia Theta | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000007 | Encyclopedia Omega | John Adams | Pub 1 | Harvard | 05/03/2016 |
| 1000000000009 | The School Bus | Mary Jane | Pub 1 | Soul | 05/03/2016 |
| 1000000000011 | Data Structures | Alan Turing | Pub 1 | Internet | 05/03/2016 |
| 1000000000013 | Alphabet Soup | Joseph Stalin | Pub 1 | Princeton | 05/03/2016 |
| 1000000000016 | Universal History | John Adams | Pub 1 | Princeton | 05/03/2016 |
| 1000000000018 | Computer Arch | Matthew Warshaw | Pub 1 | City | 05/03/2016 |
| 1000000000020 | I am the Wizard Now | Harry Potter | Pub 1 | Hogwarts | 05/03/2016 |

*Reader Search Books by Publisher*




*Reader Search Books by Author*




*Reader View Account*