

**Team Project:**  
**Deliverable 3 – Project Phase 1**  
**CSCE 5430 (Spring 2023)**

**Project Title:**  
**Online Library Management System**

**Group Name:**  
**Penguins**

**Group Members:**

No.	Name	Student ID	Email
1	Ashraf Syed	11618668	<a href="mailto:AshrafSyed@my.unt.edu">AshrafSyed@my.unt.edu</a>
2	Padmini Kuchukulla	11605145	<a href="mailto:KuchukullaPadminiLNU@my.unt.edu">KuchukullaPadminiLNU@my.unt.edu</a>
3	Abhijith Reddy Mandagiri	11603247	<a href="mailto:AbhijithReddyMandagiri@my.unt.edu">AbhijithReddyMandagiri@my.unt.edu</a>
4	Deepna Thalanki	11644830	<a href="mailto:deepnathalanki@my.unt.edu">deepnathalanki@my.unt.edu</a>
5	Raghu Vamsi Kondapalli	11663360	<a href="mailto:raghuvamsikondapalli@my.unt.edu">raghuvamsikondapalli@my.unt.edu</a>
6	Vamsi Venkat Manepalli	11656561	<a href="mailto:Vamsivenkatmanepalli@my.unt.edu">Vamsivenkatmanepalli@my.unt.edu</a>
7	Sai Vishwak Jadala	11673439	<a href="mailto:saivishwakjadala@my.unt.edu">saivishwakjadala@my.unt.edu</a>

### **3.Report**

#### **a.Requirements**

##### **FUNCTIONAL REQUIREMENTS:**

###### **Landing Page:**

###### **Description of feature:**

Gives the user a choice, whether to join as an admin or a student as per credentials.

###### **Functional Requirements :**

On choosing the respective options, each of the buttons redirects the user to the login page required for accessing the library system for utilization.

###### **New User Registration:**

###### **Description of feature:**

All users i.e students in this context have the option to use this feature to provide their contact information when creating accounts. This information includes name, email address, and contact number etc as part of the registration process.

###### **Functional requirements**

Possible feature for users to protect their account privacy is the ability to create strong passwords. This requirement allows for safe browsing and secure use of the system for students and protects their privacy.

The system must be able to validate data and establish whether all of the data was provided by the user, among other functional requirements. This mode will be more beneficial with the backend part of the website being embedded with the front end. Which will happen in the deliverables to come.

###### **User Login Page:**

###### **Description of feature:**

The user logs into the system using this function. Before logging on, they must input their user ID and password for authentication. The user cannot access the system if they are invalid.

If a user forgets their password, the system should offer a mechanism to reset it. This is accomplished by sending a reset code to their registered cellphone number. For this deliverable we will be preparing only the front end pages that will be used for all the

above mentioned functionalities. The implementation will be possible in the upcoming phases of the project.

### **Functional requirements**

When a user registers, they supply a user id(similar to that of student id). To access the system, the user must identify themselves with a valid user id and password. The system then executes an authorization procedure to determine what user level has access.

The user must have the capacity to log out after they are done using the system. The system should restrict access to specific features or data based on the user's role in the library system, such as a librarian or a user.

### **Search Book Page:**

#### **Description of feature**

Both admins and students have access to this function. We may search for books using the book's ID, name, category, or author. But each type of user has different use cases with the search feature of the website.

#### **Functional requirements**

The database must be searchable by the system using a chosen search type.

### **Admin User:**

### **Register New Book:**

#### **Description of feature**

With this feature, the administrator may include new books in the library. Based on the necessities of the students, the admin has this feature to add new books to the system for expanding the horizon of the options for the students.

#### **Functional requirements**

Information verification capabilities for systems are required. Only authenticated users are allowed to access this page and use the feature for the required purposes.

### **View Registered Students:**

#### **Description of feature:**

The admin user has a page to view a report of the students who have registered with the system. This allows the admin to keep track of all the interactions a given student is having with the system and maintain stability of the library inventory.

### **Functional requirements**

The page will have a search feature that enables the admin to track a particular student using their name or student ID number. This allows the admin to check for deadlines and potential dues that the student owes to the library

### **Student User:**

#### **View Books Issued to the user :**

##### **Description of feature**

The system should make it possible for the student to examine their personal data, including their name, student ID, contact details, and borrowing history. Students can view the books that have been issued to them using this page.

### **Functional requirements**

If the item is not overdue and has not been requested by another user, the system should enable the student to renew it. This would allow for the student to retain the book without causing any conflicts with other requests for that particular book.

### **Performance Requirements**

As of this phase of the project, there cannot be a performance requirement as the front end alone cannot determine any quantifiable measure of the performance of the system. Hence, these requirements will be introduced in the later phases of the project.

### **Design Constraints**

#### **Web support**

Web requirement is a major support which is needed as the entire application will be loaded on to the cloud. Internet access and server maintenance play a significant role in the functionality of the project. As of this phase, the front end constraints would be version mismatch between browsers and the script code used for the front end development.

#### **Security**

This being only phase 1 of the project, the project will not be having any security measures. Since it's only a front end phase, there will not be a need for security since it's only on a local system.

## **NON-FUNCTIONAL REQUIREMENTS:**

### **Requirements Of Product :**

#### **Requirement Efficiency :**

Staff members of Library and patrons will have quick access to books due to the computerization of library operations like issuing, tracking and fine calculation etc.

#### **Reliability Requirement :**

The system will accurately be able to handle member registration, member validation, book classification, transaction and search, and fine calculation, if necessary and any other tasks of the project. The project being sophisticated and reliable becomes our major priority

#### **Demand for Usability :**

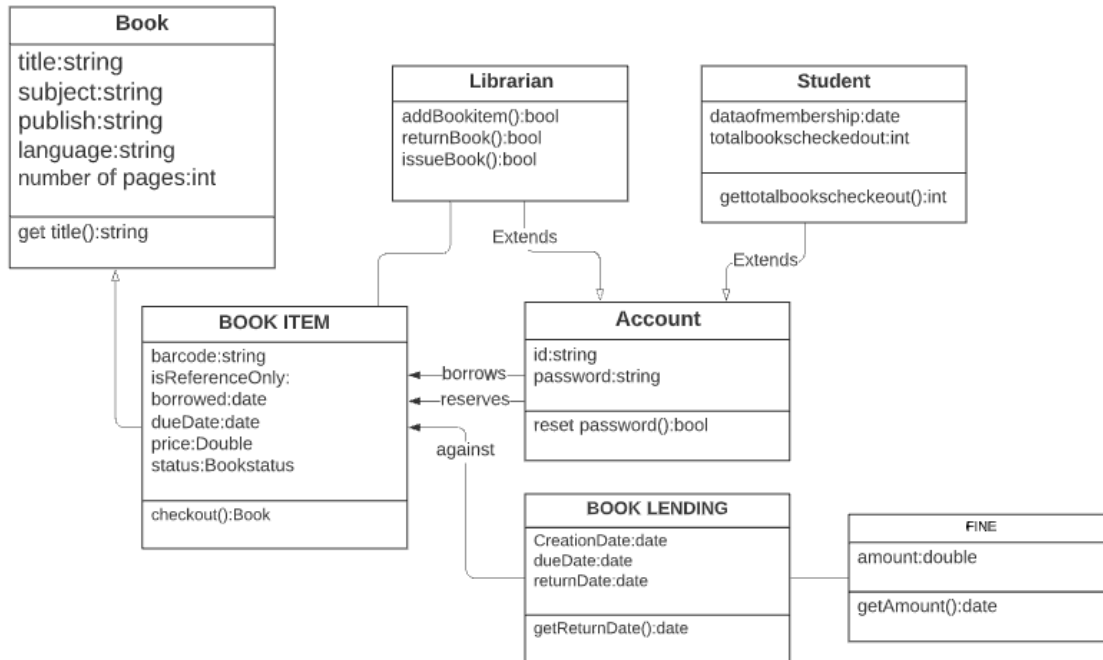
The web has been designed in a simple, sophisticated and user friendly manner. Any user, be it admin or student will find it quite easy and convenient to navigate through the site to complete any of their tasks or jobs for which they are using the site.,

#### **Scalability :**

The system will be designed keeping in mind future potential for growth and changes to the functionalities. The desire would be to keep it scalable to software growth and also to the users threshold for the website.

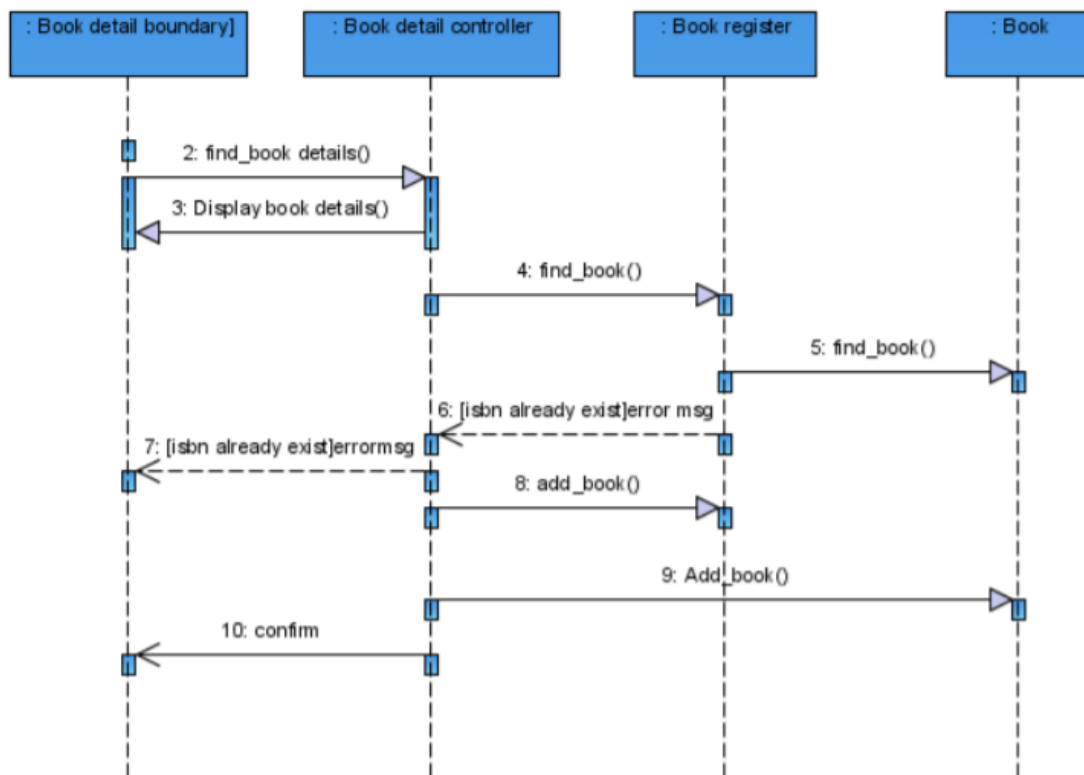
## b.UML Design

**Class diagram** - It shows the static view of the system. Here we have multiple components like book, librarian, student, book item, account, book lending and fine. Each component has multiple entities associated with it.

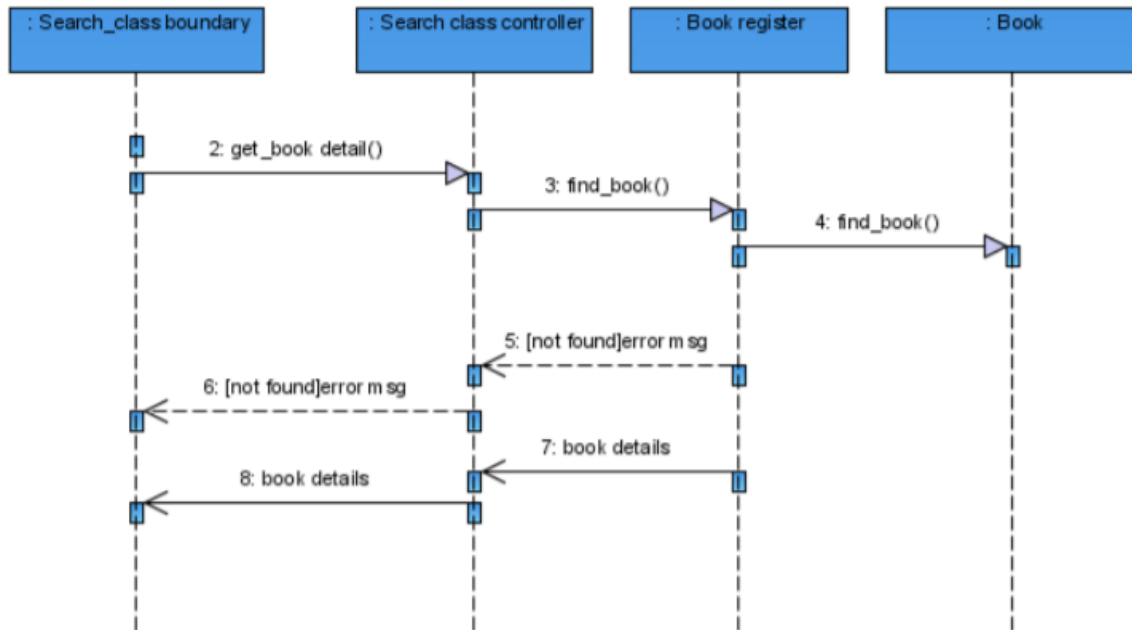


**Sequence diagram** - It shows the time sequence associated with the object interactions. Here we have multiple use cases like handling book details, search books and issue books. Below are the sequence diagrams.

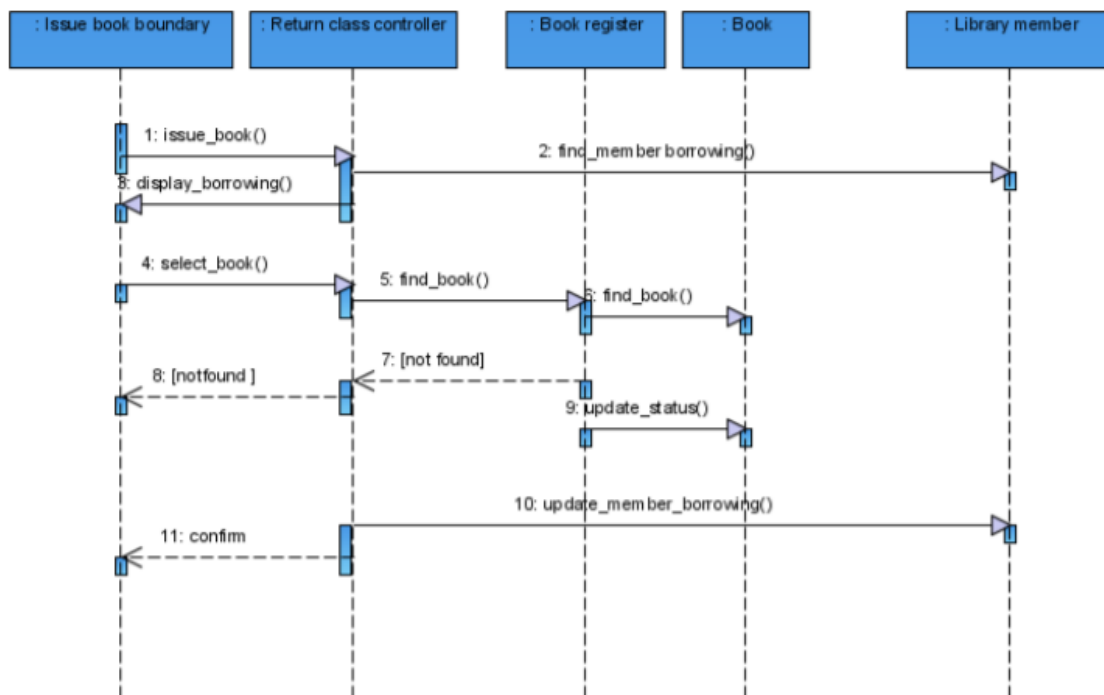
### **Handle Book Details**



## Search Books



## Issue Book





**Use case diagram** - This shows the relations between use cases and actors. Below is the diagram that depicts how a student can do multiple actions in the Online Library Management System



## c. Test Cases

### 1. Student Login Page Test Case:

The Test case script for the Student login page is

```
<script>

    const loginForm = document.getElementById("login-form");

    const loginButton =
document.getElementById("login-form-submit");

    loginButton.addEventListener("click", (e) => {

        e.preventDefault();

        const username = loginForm.username.value;

        const password = loginForm.password.value;

        if (username === "penguins" && password === "penguins") {

            alert("You have successfully logged in " + username);

            location.reload();

        } else {

            alert("Invalid Credentials");

        }

    })

</script>
```

This is an HTML code for a student login page. It uses Bootstrap to style the page and includes a form for users to input their username and password. If the user enters incorrect login credentials, an alert message will appear. The page also includes a link for users to sign up if they do not have an account. The code includes JavaScript that listens for a click event on the login button, preventing the default form submission and obtaining the values entered in the username and password fields.

The Above Script is to validate the Login page for the student and to check whether the page is working as intended. As it is a Static page the username and password are set to “**penguins**” and “**penguins**” to test. Further, in the dynamic page, the username and password are retrieved from the backend server and validated to login.

The Below screenshot shows the page response when an invalid username and password are entered. Thus this shows the functionality of the test script.

When the correct inputs that is “**penguins**” and “**penguins**” as username and password are given to the page the page response is as below.

This shows the successful validation of the user and sends an alert message of the successful login.

## 2. Admin Login Page Test Case:

The Test Case script for the Admin login page is

```
<script>

    const loginForm = document.getElementById("login-form");

    const loginButton =
    document.getElementById("login-form-submit");

    loginButton.addEventListener("click", (e) => {

        e.preventDefault();

        const username = loginForm.username.value;

        const password = loginForm.password.value;

        if (username === "admin" && password === "admin") {

            alert("You have successfully logged in.");

            window.location.href = "afterlogin.html";

        } else {

            alert("Invalid username or password. Please
try again.");

        }

    })

}
```

```
    });  
  
</script>
```

This code is a JavaScript function that handles a login form submission. It checks if the entered username and password match the expected values, and if so, it redirects the user to a page called "afterlogin.html". Otherwise, it displays an error message indicating that the username or password is invalid. As a test case, one could try entering a valid username and password and verify that the function correctly logs in the user and redirects them to the desired page. Additionally, one could try entering an invalid username or password and verify that the function correctly displays an error message and does not redirect the user to the afterlogin.html page. One could also test the behavior of the function when the login form is submitted without any input or with incomplete input, and verify that it behaves correctly in these cases as well.

The Admin Login page response when invalid input is given as shown below. This shows that the validation is working as intended.

The Admin login page works as intended with valid authentication when “**admin**” and “**admin**” as input for the username and password in their respective fields. The page response is shown in the snippet below.

On successful authentication the page redirects to the afterlogin.html User interface.

## d. User Manual

Since our project is designing a web page for Online Library Management System, the end user is provided with the following link

<http://127.0.0.1:8000/>, which navigates to our web page.

The home page of our website for the end user looks like

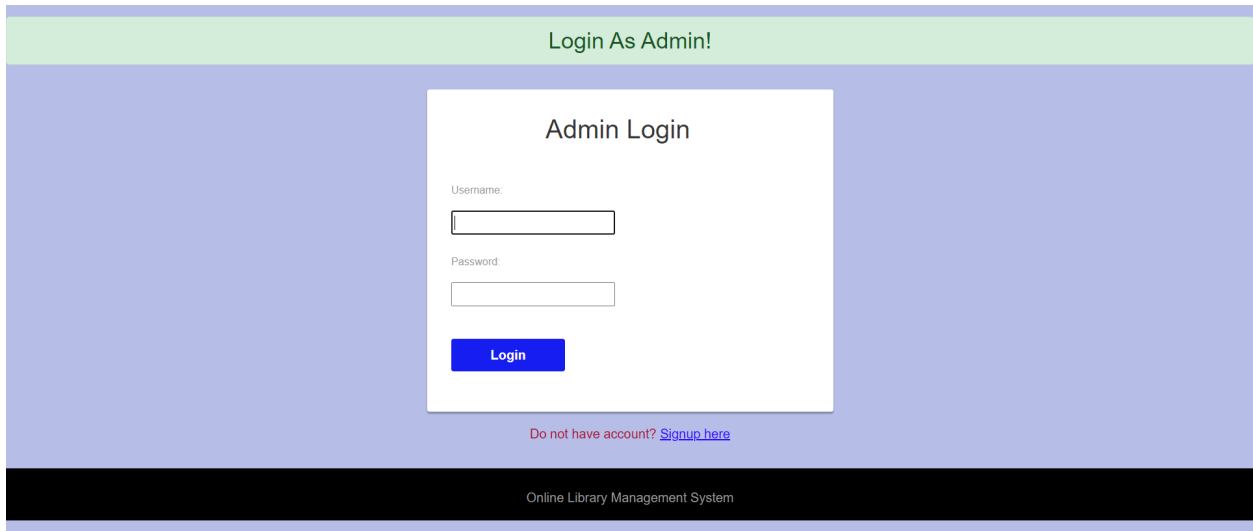
As he/she scrolls down, will be able to see the following, where there are two different portals i.e, Admin and Student.

Now, if he/she wants to enter into the Admin portal, then he/she can either click the Admin button which is in blue color or directly the Admin button which is on the top left corner of the screen. Then, he/she can view the following

If the user is using our website for the first time, he/she can click on the signup. This redirects the user to the admin signup page, which looks like the following

Provide the required details for signing up as shown below.

After giving details, one can click on the Sign Up button. Then, it redirects to the login page as below.



Login As Admin!

Admin Login

Username:

Password:

Login

Do not have account? [Signup here](#)

Online Library Management System

Give the details as given in the previous page to login as admin.

The image shows a web interface for an 'Online Library Management System'. At the top, a light green banner contains the text 'Login As Admin!'. Below this, a white rectangular box is centered on a light blue background. The box is titled 'Admin Login' and contains two input fields: 'Username:' with the value 'rk1997' and 'Password:' with masked characters '\*\*\*\*\*'. A blue 'Login' button is positioned below the password field. At the bottom of the white box, there is a red link that says 'Do not have account? [Signup here](#)'. The entire page is framed by a black footer bar at the bottom with the text 'Online Library Management System'.

After successful login, one can view what all the functionalities can be performed by admin as below.

Now, to get into the student portal click on the logout button on the top right corner of the screen, then one redirects to the home page as below.

Now the user can either click the Student button which is in blue color or directly the Student button which is on the top left corner of the screen. Then, he/she can view the following screen.

If the student is using our website for the first time, he/she can click on the signup. This redirects the user to the student signup page, which looks like the following.

Provide the required details for signing up as shown below.

After giving details, one can click on the Sign Up button. Then, it redirects to the login page as below.

This screenshot shows the 'Student Login' interface. At the top, a green banner reads 'Login As Student!'. Below it, a white login box contains the title 'Student Login', a 'Username:' label with a text input field containing 'sn3110', a 'Password:' label with a masked password input field, and a blue 'Login' button. Below the login box, a red link says 'Do not have account? [Signup here](#)'. The footer is a black bar with the text 'Online Library Management System'.

If credentials don't match the previous page, then there comes a pop up as shown below.

This screenshot shows the same 'Student Login' interface as the previous one, but with an error. The 'Username' field still contains 'sn3110', but the 'Password' field is empty. A red error message, 'Incorrect Password', is displayed below the password field. The rest of the page, including the 'Login' button and the footer, remains the same.

After successful login, one can view what all the functionalities can be performed by the student as below.

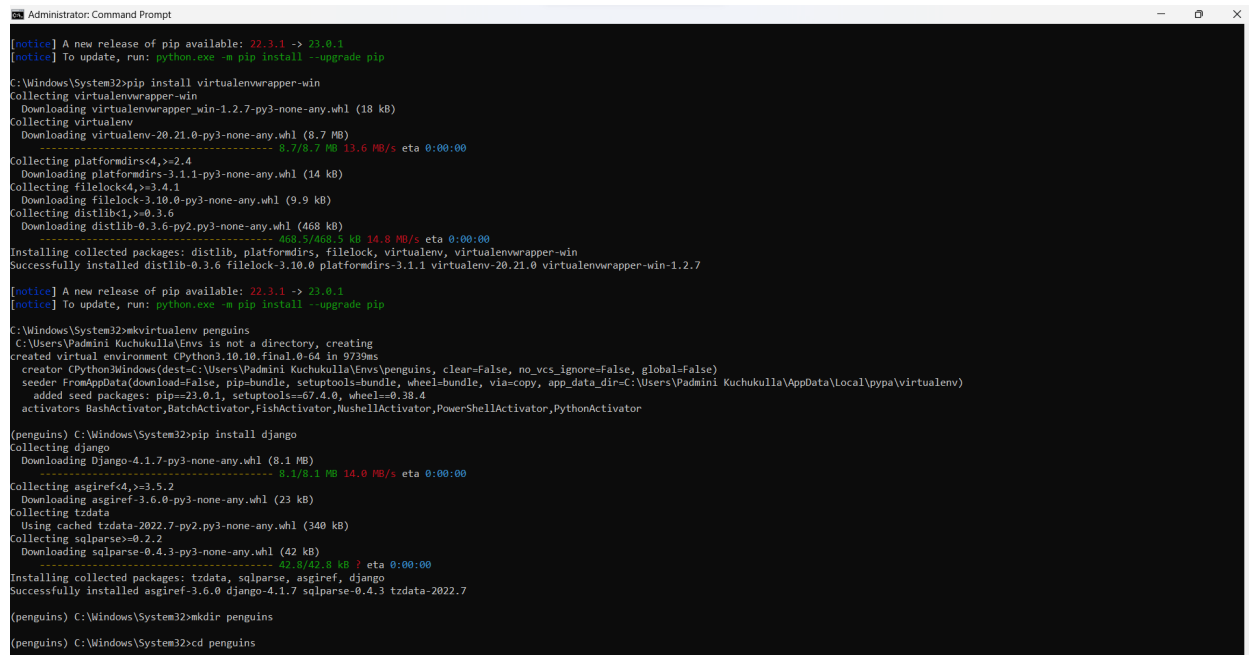
```
mkvirtualenv penguins
```



As soon as we create the virtual environment named “penguins”, we will start working on the environment.

Now, in the environment we have installed django using the following command.

```
pip install django
```



```
Administrator: Command Prompt

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Windows\System32>pip install virtualenvwrapper-win
Collecting virtualenvwrapper-win
  Downloading virtualenvwrapper-win-1.2.7-py3-none-any.whl (18 kB)
Collecting virtualenv
  Downloading virtualenv-20.21.0-py3-none-any.whl (8.7 MB)
----- 8.7/8.7 MB 13.6 MB/s eta 0:00:00
Collecting platformdirs<4,>=2.4
  Downloading platformdirs-3.1.1-py3-none-any.whl (14 kB)
Collecting filelock<4,>=3.4.1
  Downloading filelock-3.10.0-py3-none-any.whl (9.9 kB)
Collecting distlib<1,>=0.3.6
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
----- 468.3/468.3 kB 14.8 MB/s eta 0:00:00
Installing collected packages: distlib, platformdirs, filelock, virtualenv, virtualenvwrapper-win
Successfully installed distlib-0.3.6 filelock-3.10.0 platformdirs-3.1.1 virtualenv-20.21.0 virtualenvwrapper-win-1.2.7

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Windows\System32>mkvirtualenv penguins
C:\Users\Padmini Kuchukulla>Envs is not a directory, creating
created virtual environment CPython3.10.10.final.0-64 in 9739ms
creator CPython3Windows(dest=C:\Users\Padmini Kuchukulla>Envs\penguins, clear=False, no_vcs_ignore=False, global=False)
reeder fromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\Padmini Kuchukulla\AppData\Local\pypa\virtualenv)
added seed packages: pip==23.0.1, setuptools==67.4.0, wheel==0.38.4
activators BashActivator,BatchActivator,FishActivator,MsShellActivator,PowerShellActivator,PythonActivator

(penguins) C:\Windows\System32>pip install django
Collecting django
  Downloading Django-4.1.7-py3-none-any.whl (8.1 MB)
----- 8.1/8.1 MB 14.0 MB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting tzdata
  Using cached tzdata-2022.7-py2.py3-none-any.whl (340 kB)
Collecting sqlparse<0.2.2
  Downloading sqlparse-0.4.3-py3-none-any.whl (42 kB)
----- 42.0/42.0 kB 4.0 MB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.6.0 django-4.1.7 sqlparse-0.4.3 tzdata-2022.7

(penguins) C:\Windows\System32>mkdir penguins
(penguins) C:\Windows\System32>cd penguins
```

Now, after successful installation of django, we created the directory named “penguins” in the virtual environment using the following command

```
mkdir penguins
```

Now, change the path to the newly created directory using the following command

```
cd penguins
```

Now, in this directory we start our project using the django framework command which is as follows

```
django-admin startproject OnlineLibraryMangementSystem
```

After successful creation of the project directory, change the working path to this directory using the following command

```
cd OnlineLibraryMangementSystem
```

Now, to get the webpage link, we used python command which is as follows

```
python manage.py runserver
```

The manage python file is created by django in default when we start the project.

In the output we can see our website link as <http://127.0.0.1:8000/>

```
Administrator: Command Prompt
Using cached tzdata-2022.7-py2.py3-none-any.whl (340 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.3-py3-none-any.whl (42 kB)
-----
42.8/42.8 KB | eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.6.0 django-4.1.7 sqlparse-0.4.3 tzdata-2022.7

(penguins) C:\Windows\System32>mkdir penguins
(penguins) C:\Windows\System32>cd penguins
(penguins) C:\Windows\System32\penguins>django-admin startproject
usage: django-admin startproject [-h] [--template TEMPLATE] [--extension EXTENSIONS] [--name FILES] [--exclude [EXCLUDE]] [--version] [-v {0,1,2,3}] [--settings SETTINGS] [--pythonpath PYTHONPATH]
                                   [--traceback] [--no-color] [--force-color]
                                   name [directory]
django-admin startproject: error: You must provide a project name.

(penguins) C:\Windows\System32\penguins>django-admin startproject OnlineLibraryManagementSystem
(penguins) C:\Windows\System32\penguins>cd OnlineLibraryManagementSystem
(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 19, 2023 - 13:58:30
Django version 4.1.7, using settings 'OnlineLibraryManagementSystem.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[19/Mar/2023 13:58:53] "GET / HTTP/1.1" 200 10681
[19/Mar/2023 13:58:53] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[19/Mar/2023 13:58:53] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[19/Mar/2023 13:58:53] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[19/Mar/2023 13:58:53] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
Not Found: /favicon.ico
[19/Mar/2023 13:58:53] "GET /favicon.ico HTTP/1.1" 404 2133

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py startapp accounts
(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>
```

Now, when we open the link, we can just see that installation is successful.

Then, download and install postgresql, here we downloaded the 14.7 version using the following link

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

After successful installation of postgresql, now download and install pgadmin4, here we downloaded the 6.19 version using the following link

<https://www.postgresql.org/ftp/pgadmin/pgadmin4/v6.19/windows/>

Now, we created a folder named “templates”, which contains all the html files required for our project, which are coded separately by all of our teammates.

Then, we also created a folder named “static”, which contains another folder inside it named “images”, which contains all the images we used in developing the frontend.

While we are setting up these, it asks for a password, which we gave it as “Penguins”. This should be updated in the settings.py file, which is created by django in default.

After this step, we will try to run the server using the following command

```
python manage.py runserver
```

```
Administrator Command Prompt - python manage.py runserver
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32\workon penguins
(penguins) C:\Windows\System32>cd penguins

(penguins) C:\Windows\System32\penguins>cd OnlineLibraryManagementSystem

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\db\backends\postgresql\base.py", line 24, in <module>
    import psycopg2 as Database
ModuleNotFoundError: No module named 'psycopg2'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 953, in run
    self.target(*self.args, **self.kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\commands\runserver.py", line 125, in inner_run
    autoreload.raise_last_exception()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 87, in raise_last_exception
    raise exception[1]
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\_init_.py", line 398, in execute
    autoreload.check_errors(django.setup())
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\_init_.py", line 24, in setup
    apps.populate(settings.INSTALLED_APPS)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\apps\registry.py", line 116, in populate
    app_config.import_models()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\apps\config.py", line 269, in import_models
    self.models_module = import_module(models_module_name)
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\importlib\_init_.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1059, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 883, in exec_module
  File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\contrib\auth\models.py", line 3, in <module>
    from django.contrib.auth.base_user import AbstractBaseUser, BaseUserManager
```

Now if all the required modules were installed, we would be able to see the frontend working fine on our webpage. But we can see the `ModuleNotFoundError`, which says there is no “psycopg2” module. This module is used as the database connecting adapter to postgresql. So, now we install this module using the following command.

```
pip install psycopg2
```

```
Administrator Command Prompt - python manage.py runserver

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>pip install psycopg2
Collecting psycopg2
  Downloading psycopg2-2.9.5-cp310-cp310-win_amd64.whl (1.2 MB)
    ----- 1.2/1.2 MB 5.7 MB/s eta 0:00:00
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.5

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

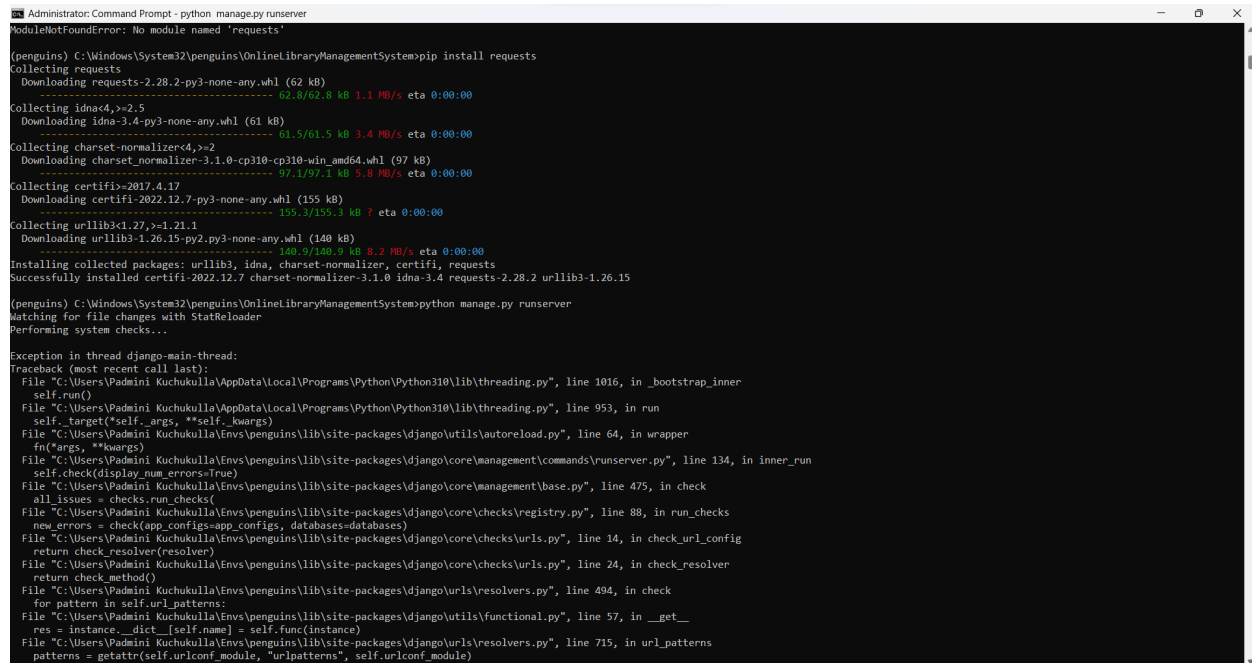
Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 953, in run
    self.target(*self.args, **self.kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run
    self.check(display_num_errors=True)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\base.py", line 475, in check
    all_issues = checks.run_checks(
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\registry.py", line 88, in run_checks
    new_errors = check(app_configs=app_configs, databases=databases)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 14, in check_url_config
    return check_resolver(resolver)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 24, in check_resolver
    return check_method()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 494, in check
    for pattern in self.url_patterns:
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\functional.py", line 57, in _get__
    res = instance._dict[self.name] = self.func(instance)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 715, in url_patterns
    patterns = getattr(self.urlconf_module, "urlpatterns", self.urlconf_module)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\functional.py", line 57, in _get__
    res = instance._dict[self.name] = self.func(instance)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 708, in urlconf_module
    return import_module(self.urlconf_name)
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\importlib\_init_.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1059, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
  File "<frozen importlib._bootstrap>", line 1059, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1004, in _find_and_load_unlocked
```

After this step, again we will try to run the server using the following command

```
python manage.py runserver
```

We can see the `ModuleNotFoundError`, which says there is no “requests” module. This module is used for handling all the http requests. So, now we install this module using the following command.

```
pip install requests
```



```
Administrator: Command Prompt - python manage.py runserver
ModuleNotFoundError: No module named 'requests'

(penguins) C:\Windows\System32\penguins\OnlineLibrary\ManagementSystem>pip install requests
Collecting requests
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
-----
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
-----
Collecting charset-normalizer<4,>=2
  Downloading charset-normalizer-3.1.0-cp310-cp310-win_and64.whl (97 kB)
-----
Collecting certifi>=2017.4.17
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
-----
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.15-py2.py3-none-any.whl (140 kB)
-----
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2022.12.7 charset-normalizer-3.1.0 idna-3.4 requests-2.28.2 urllib3-1.26.15

(penguins) C:\Windows\System32\penguins\OnlineLibrary\ManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 953, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run
    self.check(display_num_errors=True)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\base.py", line 475, in check
    all_issues = checks.run_checks(
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\registry.py", line 88, in run_checks
    new_errors = check(app_configs=app_configs, databases=databases)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 14, in check_url_config
    return check_resolver(resolver)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 24, in check_resolver
    return check_method()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 494, in check
    for pattern in self.url_patterns:
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\functional.py", line 57, in __get__
    res = instance.__dict__[self.name] = self.func(instance)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 715, in url_patterns
    patterns = getattr(self.urlconf_module, "urlpatterns", self.urlconf_module)
```

After this step, again we will try to run the server using the following command

```
python manage.py runserver
```

We can see the `ModuleNotFoundError`, which says there is no “bs4” module. This module is beautifulsoup4, which is used for web scraping. So, now we install this module using the following command.

```
pip install bs4
```

```
Administrator Command Prompt - python manage.py runserver
from bs4 import BeautifulSoup
ModuleNotFoundError: No module named 'bs4'

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>pip install bs4
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
    Preparing metadata (setup.py) ... done
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.11.2-py3-none-any.whl (129 kB)
----- 129.4/129.4 kB 1.5 MB/s eta 0:00:00
Collecting soupsieve>1.2
  Downloading soupsieve-2.4-py3-none-any.whl (37 kB)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) ... done
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1264 sha256=511b117a6b4843304b058e4e11078a9594491a1ff27361d02e44a0fecafcae61
  Stored in directory: c:\users\padmini\kuchukulla\appdata\local\pip\cache\wheels\25\42\45\b773edc52acb16cd2db4cf1a0b47117e2f69bb4eb30ed0be70
Successfully built bs4
Installing collected packages: soupsieve, beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.11.2 bs4-0.0.1 soupsieve-2.4

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\threading.py", line 953, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run
    self.check(display_num_errors=True)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\management\base.py", line 475, in check
    all_issues = checks.run_checks(
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\registry.py", line 88, in run_checks
    new_errors = check(app_configs=app_configs, databases=databases)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 14, in check_url_config
    return check_resolver(resolver)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\core\checks\urls.py", line 24, in check_resolver
    return check_method()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 494, in check
    for pattern in self.url_patterns:
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\functional.py", line 57, in __get__
    res = instance.__dict__[self.name] = self.func(instance)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\urls\resolvers.py", line 715, in url_patterns
    patterns = getattr(self.urlconf_module, "urlpatterns", self.urlconf_module)
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\functional.py", line 57, in __get__
```

After this step, again we will try to run the server using the following command

```
python manage.py runserver
```

We can see the ModuleNotFoundError, which says there is no “speech\_recognition” module. This module is used for recognizing speech, which we will be using for voice-based search. So, now we install this module using the following command.

```
pip install SpeechRecognition
```

```
Administrator: Command Prompt - python manage.py runserver
ModuleNotFoundError: No module named 'speech_recognition'

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>pip install SpeechRecognition
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.10.0-py2.py3-none-any.whl (32.8 MB)
    -----
    32.8/32.8 MB 11.9 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.26.0 in c:\users\padmini kuchukulla\envs\penguins\lib\site-packages (from SpeechRecognition) (2.28.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\padmini kuchukulla\envs\penguins\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (3.1.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\padmini kuchukulla\envs\penguins\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\padmini kuchukulla\envs\penguins\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\padmini kuchukulla\envs\penguins\lib\site-packages (from requests>=2.26.0->SpeechRecognition) (3.4)
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.10.0

(penguins) C:\Windows\System32\penguins\OnlineLibraryManagementSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
accounts.Book: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
         HINT: Configure the DEFAULT_AUTO_FIELD setting or the AccountsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
accounts.IssuedBook: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
         HINT: Configure the DEFAULT_AUTO_FIELD setting or the AccountsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
accounts.StudentExtra: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
         HINT: Configure the DEFAULT_AUTO_FIELD setting or the AccountsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 3 issues (0 silenced).
Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\django\db\backends\base\base.py", line 282, in ensure_connection
    self.connect()
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\django\utils\asyncio.py", line 26, in inner
    return func(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\django\db\backends\base\base.py", line 263, in connect
    self.connection = self.get_new_connection(conn_params)
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\django\utils\asyncio.py", line 26, in inner
    return func(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\django\db\backends\postgresql\base.py", line 215, in get_new_connection
    connection = Database.connect(**conn_params)
  File "C:\Users\Padmini Kuchukulla\Env\penguins\lib\site-packages\psycopg2\_init_.py", line 122, in connect
    conn = _connect(dsn, connection_factory=connection_factory, **kwargs)
psycopg2.OperationalError: connection to server at "localhost" (::1), port 5432 failed: FATAL:  password authentication failed for user "postgres"

The above exception was the direct cause of the following exception:
Traceback (most recent call last):
```

After this step, again we will try to run the server using the following command  
`python manage.py runserver`

## f. Peer review feedback

During the productive peer review session, the team engaged in an open and collaborative discussion, sharing their thoughts and concerns about the project and identifying ways to overcome hurdles. The team focused on the front-end parts of the project, specifically the user interface, to improve its user-friendliness and efficiency.

In the review session, the team received feedback to improve the login process by creating separate pages for admin and user. To implement this suggestion, the team decided to include an interactive page with various options for admin and user features, instead of just a navigation bar.

As part of the improvement process, the team also planned to incorporate alerts to keep the users informed about the status of the pages which was pointed out in the peer review session. This would help them stay updated and aware of any changes or updates to the system. Moreover, the team made the page more responsive to user actions, ensuring that the users could navigate the pages smoothly and easily. As per the suggestions regarding the backend implementation that were raised will be used in the phase 2 implementation of the project.

In summary, the team received valuable feedback during the review session, which they implemented with add-ons like an interactive page and alerts to improve the login process. The changes were aimed at making the system more user-friendly and efficient. The team also made the page responsive to user actions to enhance the overall user experience.

### **g. Reflection**

In Phase 1, we have accomplished the majority of the tasks that we had initially planned for. The project required extensive analysis and design, which took up a considerable amount of time. Once that was completed, we distributed the tasks among our team members and began coding. In this phase, we focused solely on the front end, using HTML and other front-end technologies for the web pages. Fortunately, the spring break allowed us to invest more time and effort into the project, allowing us to complete all of our objectives.

We are particularly proud of our team's collaboration and communication. We regularly connected and delegated tasks among ourselves, which enabled us to reach our targets efficiently. The testing for the front end pages was difficult but we faced this situation by using static web pages to test the pages. Moreover, constant involvement from the entire team significantly reduced code redundancy, which was a crucial improvement for the project. Looking forward, we believe that our successful completion of Phase 1 will set us up well for the next phases of the project.

#### **Member's Contribution:**

<b>Member name</b>	<b>Contribution description</b>	<b>Overall Contribution (%)</b>	<b>Note (if applicable)</b>
Ashraf Syed	Requirements, Meeting minutes updation, involved in building the code	14.3%	
Padmini Kuchukulla	Contributed in pushing code to github, user manual, code compilation, involved in building the code	14.3%	
Abhijith Reddy Mandagiri	Contributed in Test cases, Reflection, Review feedback	14.3%	

	from peer Evaluation, involved in building the code		
Deepna Thalanki	Contributed in UML diagrams, involved in building the code	14.3%	
Raghu Vamsi Kondapalli	Contributed in UML diagrams, involved in building the code, Meeting minutes updation	14.3%	
Vamsi Venkat Manepalli	Involved in core functionalities document, building the code	14.3%	
Sai Vishwak Jadala	Involved in core functionalities document, building the code	14.3%	