Team Project:
Deliverable 5 – Final Project
CSCE 5430 (Spring 2023)

Project Title:
Online Library Management System

Group Name:
Penguins

Group Members:

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

2.Report

a.Requirements

The phase 2 of the project development plans have changed from our initial plan. We have changed the plan from developing only Admin features for phase 2 to including both major features of student and admin to phase 2. But not all features are added, only the common pages with similar page structure for both admin and student are included in this phase. The reason for changing the initial plan is that, after the previous code inspection session with the Team Achievers, based on the suggestions they have given to our previous phase, we believe keeping a buffer scope in our final phase will give us a good opportunity to incorporate all the suggestions and make the project to the best of our abilities. The updated phase 3 will be the addition of features like voice-based search, Automatic categorization, payment gateway and implementing any recommended changes or updates given during the peer evaluation.

FUNCTIONAL REQUIREMENTS:

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:

The system must be able to validate data and establish whether all the data was provided by the user, among other functional requirements. This would also provide the much-needed security and authenticity to the user login.

User Login:

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.

Functional requirements:

When a user registers, they supply a user id (like 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.

Functional requirements:

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.

Admin User:

Register New Book:

Description of feature:

With this feature, the administrator alone can have the access to include new books in the library. Based on 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.

Functional requirements:

The database table must be able to accept entries from the system for book information and copy count. The entry of the new book must follow all the proper schema set by the database for proper inclusion.

Functional requirements:

The system needs to be able to prevent two books from sharing the same book id. The primary key of the database does not allow for duplication.

Issue and Returned Books:

Description of feature:

This function enables the administrator to issue and receive books as well as see reports on books issued. It’s an exclusive feature for the admin, which allows them to manage the inventory of the library by keeping track of the books being checked out and checked in.

Functional requirements:

The number of books must be updated by the system, each time a book is checked out, it's the admins job to assure that the database is updated for every transaction taking place. Just to make sure the database doesn't fall into redundancy.

Functional requirements:

The system must be able to verify that the book is available before distributing it. Book would only be issued to a student, if there is a copy of the book available in the database that hasn't already been issued.

Functional requirements:

Information about the issue and return dates, with the return date set to 15 days. The system should set a timer and update it daily to back track to the finish of 15 days limit.

Functional requirements:

The system should set due dates for loaned books and notify users through email or their mobile device when they are past due. The above requirement will help in assisting to this feature of notifying the student.

Functional requirements:

The system should keep track of returned books and make sure they are appropriately added back to the database count of the library system.

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 book.

Performance Requirements

Response Time - The system must react quickly and effectively as users look for books, borrow and return books, and manage their accounts. The communication between the front end and the backend must be precise and fast.

Availability- The system should be available 24/7 so that users can utilize it whenever they want. As little downtime as possible should be built into the system, and any upgrades or maintenance should be scheduled for off-peak hours.

Scalability - To handle an increase in users, transactions, and data without impacting its functionality, the system must be scalable. To support future development, hardware and software should be updated in accordance with the architecture of the system.

Security - To handle an increase in users, transactions, and data without impacting its functionality, the system must be scalable. To support future development, hardware and software should be updated in accordance with the architecture of the system.

Usability - The user interface must be straightforward, easy to use, and uncomplicated. These requirements must be taken into consideration while designing the system's user interface, which should also provide clear, comprehensive instructions for using it.

Design Constraints

Web support:

As the complete program will be placed onto the cloud, a crucial support that is required is the web. The project's functionality depends heavily on having access to the internet and maintaining the servers.

Language support:

English being the well-known famous language, the entire web-app is in English. Support for other languages might be difficult to achieve.

Security:

This being the major phase of the project development, the project has been developed with extreme security in mind and the best experience for the user in mind.

Other Requirements:

Internal Interface Requirements:

An internal interface for the online computerization of library activities is frequently made up of a collection of software packages or modules designed to manage various aspects of library operations. The following are some crucial components that could be present: Using the cataloging module, library employees can create, maintain, and update the catalog of books and other things. It frequently offers capabilities for filtering, sorting, and catalog searching. The circulation module allows for the lending and borrowing of library materials. It can be used by the staff to control fines and fees, maintain tabs on material issues and due dates, and check items in and out.

Reports Module:

Users have access to reports and information on library operations using this module. There may be statistics on catalog usage, circulation, and other important factors. Databases, books, and journals are just a few of the electronic resources the library manages with the help of this module. It might be able to measure usage, control access, authenticate users, and report on that usage.

NON-FUNCTIONAL REQUIREMENTS:

Requirements Of Product:

Requirement Efficiency:

Staff members of the 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.

Performance:

While managing large volumes of data and users via the system, there shouldn't be any latency or delay. Any transactions occurring in the system will have the same experience for every user making sure they have a good and positive experience.

Demand for Usability:

The web has been created in a way that is straightforward, sophisticated, and user-friendly. Any user, whether an administrator or a student, will find it incredibly simple and comfortable to browse about the website in order to finish any activities or occupations for which they are utilizing it.

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 to the user's threshold for the website.

Organizational Requirements:

Implementation Requirements:

The complete system is developed using Python Django as the back end and HTML as the front end. Database storage will be handled by PostgreSQL, while database connectivity will be handled by the Django framework along with the unit tests for the project.

b.UML Design

UML

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

Success Diagram

Error Case Diagram

c. Test Cases

For our project we are using the Django Framework for the unit test cases. They are as follows:

Student Signup

1) We are using Django Framework

2) This is a unit test case for the student signup page where we have included characters like username, email, password, firstname, lastname, enrollment, branch.

```
response = self.client.post(self.url, data)

        self.assertEqual(response.status_code, 302)  # redirect
to studentlogin

        self.assertEqual(User.objects.count(), 1)

        user = User.objects.first()

        self.assertEqual(user.username, 'dg0107')

        self.assertEqual(user.email,
'deepthipasamgnana@gmail.com')

        self.assertTrue(user.check_password('Deepthi@0107'))

        self.assertEqual(user.first_name, 'Deepthi')

        self.assertEqual(user.last_name, 'Gnana')

        self.assertEqual(user.enrollment, 'Masters')

        self.assertEqual(user.branch, 'CS')


self.assertTrue(user.groups.filter(name='STUDENT').exists())
```

Admin Signup

1) We are using Django Framework

2) This is an unit test case for admins signup page where we have included characters like username, email, password, firstname, lastname, enrollment, branch.

```
response = self.client.post(self.url, data)

        self.assertEqual(response.status_code, 302)  # redirect
to adminlogin

        self.assertEqual(User.objects.count(), 1)

        user = User.objects.first()

        self.assertEqual(user.username, 'snr3110')

        self.assertEqual(user.firstname, 'Sreeja')

        self.assertEqual(user.lastname, 'Nanyala')

        self.assertTrue(user.check_password('Sreeja@2001'))


self.assertTrue(user.groups.filter(name='ADMIN').exists())
```

Admin Login

1) We are using Django Framework

2) We are providing username and password, if they're true and then we hit the submit then the Admin gets logged in successful.

3) Input characters of username and password are sent to pushbutton(Login) which analyzes whether it is true or false and allows Admin to get logged in.

```
response = self.client.post(self.url, data)

        self.assertEqual(response.status_code, 302)  # redirect
to admin home
```

```
        self.assertEqual(User.objects.count(), 1)

        user = User.objects.first()

        self.assertEqual(user.username, 'snr3110')

        self.assertTrue(user.check_password('Sreeja@2001'))
```

Student Login

1) We are using Django Framework

2) We are providing username and password, if they're true and then we hit the submit then the admin gets logged in successfully.

3) Input characters of username and password are sent to pushbutton(Login) which analyzes whether it is true or false and allows the admin to get logged in.

```
response = self.client.post(self.url, data)

        self.assertEqual(response.status_code, 302)  # redirect
to student home

        self.assertEqual(User.objects.count(), 1)

        user = User.objects.first()

        self.assertEqual(user.username, 'dg0107')

        self.assertTrue(user.check_password('Deepthi@0107'))
```

1) We are using Django Framework

2) If the provided details of Librarian (username and password) are incorrect i.e it does not match according to the Librarian gets signed up then the Librarian cannot get logged in.

Test case for successful librarian login - test that a librarian can successfully login with valid credentials and access the librarian dashboard.

```
response = self.client.post('/login/', {'username':
'testlibrarian', 'password': 'Test@1234'})

self.assertEqual(response.status_code, 302)

self.assertRedirects(response,
'/dashboard/librarian/')
```

Borrowing a Book

Test case for borrowing a book - test that a student can successfully borrow a book from the library.

1) We are using Django Framework

2) If the provided details of Book to the librarian if the book is available in the library then it will show that you can successfully borrow a book

```
response =
self.client.post('/dashboard/student/borrowbook/',
{'book_id': book.id})

self.assertEqual(response.status_code, 302)

self.assertEqual(BorrowedBook.objects.count(), 1)

borrowed_book = BorrowedBook.objects.first()

self.assertEqual(borrowed_book.student, student)

self.assertEqual(borrowed_book.book, book)
```

Error Test Cases.

Student signup

1) We are using Django Framework

2) We are providing username and password, if the details entered by a student are incorrect/ False then this page doesn't allow the student to get signed in/ create an account.

```
 response = self.client.post(self.url, data)

   self.assertEqual(response.status_code, 302)  # redirect
to studentlogin

   self.assertEqual(User.objects.count(), 1)

   user = User.objects.first()

   self.assertEqual(user.username, 'dg0107')

   self.assertEqual(user.email,
'deepthipasamgnana@gmail.com')

   self.assertFalse(user.check_password('Deepthi@0108'))

   self.assertEqual(user.first_name, 'Deepthi')

   self.assertEqual(user.last_name, 'Gnana')

   self.assertEqual(user.enrollment, 'Masters')

   self.assertEqual(user.branch, 'CS')


self.assertTrue(user.groups.filter(name='STUDENT').exists())
```

Admin Signup

1) We are using Django Framework

2) We are providing username and password, if the details entered by an Admin are incorrect/ False then this page doesn't allow the Admin to get signed in/ create an account.

```
response = self.client.post(self.url, data)

    self.assertEqual(response.status_code, 302)  # redirect
to adminlogin

    self.assertEqual(User.objects.count(), 1)

    user = User.objects.first()

    self.assertEqual(user.username, 'snr3110')

    self.assertEqual(user.firstname, 'Sreeja')

    self.assertEqual(user.lastname, 'Nandyala')

    self.assertFalse(user.check_password('Sreeja@2002'))


self.assertTrue(user.groups.filter(name='ADMIN').exists())
```

Admin Login

1) We are using Django Framework

2) If the provided details of Admin(username and password) are incorrect i.e it does not match according to the Admin gets signed up then the Admin cannot get logged in.

```
response = self.client.post(self.url, data)

    self.assertEqual(response.status_code, 302)  # redirect
to admin home

    self.assertEqual(User.objects.count(), 1)
```

```
        user = User.objects.first()

        self.assertEqual(user.username, 'snr3110')

        self.assertFalse(user.check_password('Sreeja@2002'))
```

Student Login

1) We are using Django Framework

2) If the provided details of Student (username and password) are incorrect i.e it does not match according to the student gets signed up then the Student cannot get logged in.

```
     response = self.client.post(self.url, data)

        self.assertEqual(response.status_code, 302)  # redirect
to student home

        self.assertEqual(User.objects.count(), 1)

        user = User.objects.first()

        self.assertEqual(user.username, 'dg0107')

        self.assertFalse(user.check_password('Deepthi@0108'))
```

1. We are using Django Framework

2. One possible error test case for the above code could be to check if the registration process fails when an invalid email address is provided

```
response = self.client.post('/register/student/',
{'username': 'teststudent', 'password': 'Test@1234',
'confirm_password': 'Test@1234', 'first_name': 'Test',
'last_name': 'Student', 'email': 'invalid_email',
'enrollment': 'Masters', 'branch': 'CS'})

self.assertEqual(response.status_code, 200)
```

```
self.assertEqual(User.objects.count(), 0)
```

Student Login

1) We are using Django Framework
2) If the provided details of Student (username and password) are incorrect i.e it does not match according to the student gets signed up then the Student cannot get logged in.

```
    response = self.client.post(self.url, data)
        self.assertEqual(response.status_code, 302)  # redirect
to student home
        self.assertEqual(User.objects.count(), 1)
        user = User.objects.first()
        self.assertEqual(user.username, 'dg0107')
        self.assertFalse(user.check_password('Deepthi@0108'))
```

```
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
......
----------------------------------------------------------------------
Ran 6 tests in 0.006s

OK
Destroying test database for alias 'default'...
```

The above image is the output of the test cases that have been successfully executed on the local server. All the codes given above have been executed as part of unit testing on the project functional method implementation.

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.

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

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

Now, the admin can add books to the library by clicking on the add book to library button as shown below.

Now after giving the details of book, click on add button then one can view the resulting page as below.

Now, to view all the books in the library click on “click here to view book in library” button. After clicking the button one will be redirected to the following page

Book Name	Author	ISBN	Category	No of Copies	Copies Issued
Mathematics	Pragwal	427408751095	Education	2	0
Mathematics	Pragwal	75384692837409	Education	67	0
Introduction to SE	Hyunsook	57484508	Education	10	0

Now in this phase, we implemented voice based search as well as text based search and automatic categorization. As it can be seen below, there is given a text area for key-based search and mike symbol for voice based search. One can search books as per their interest.

Book Name	Author	ISBN	Category	No of Copies	Copies Issued
Mathematics	Pragwal	427408751095	Education	2	0
Mathematics	Pragwal	75384692837409	Education	67	0
Introduction to SE	Hyunsook	57484508	Education	10	0

Once you are done typing or speaking, one can click search button which is shown to view the search results, which can be depicted as below.

The other functionality of admin is to issue books available to students, which can be done by clicking the “Issue new book” button, which appears on the admin after login page. By clicking this button one can see as the following

Now, the admin can click on “issue” button to issue the mentioned book details to the mentioned student details. After clicking this button the page redirects as the following

To view the books issued, admin has to click on “click here to view issued book” butto. After clicking this button we can see following page

Now, the last functionality of the admin is that he can view all the students registered to the library by clicking on “view student” button. This can be viewed as shown 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.

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

Now, as we can see above student can view their books by clicking on “My Books” button. After clicking on this button, it redirects to the following page as below.

As the other functionality of the student, he can see all the books available in the library by clicking on “Library Books”. By clicking on this, one can see the following page.

Book Name	Author	ISBN	Category	No of Copies	Copies Issued
Mathematics	Pragwal	427408751095	Education	2	0
Mathematics	Pragwal	75384692837409	Education	67	0
Introduction to SE	Hyunsook	57484508	Education	10	0

e. Compilation Instructions

For the ease of connecting backend and frontend, we are using the django framework to build our project.

To compile/execute our project, the basic requirement is the system you are working on should have python.

If there is no Python, then download and install the latest version, here we are using Python 3.10 using the following link.
https://www.python.org/downloads/

After successful installation of Python, installed django-admin as shown below using following command

```
pip install django-admin
```

```
Administrator: Command Prompt
C:\Windows\System32>pip --version
pip 22.3.1 from C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)

C:\Windows\System32>pip install django-admin
Collecting django-admin
  Downloading django_admin-2.0.2-py2.py3-none-any.whl (7.6 kB)
Collecting django-excel-response2>=3.0.0
  Downloading django_excel_response2-3.0.5-py2.py3-none-any.whl (4.8 kB)
Collecting django-six
  Downloading django_six-1.0.5-py2.py3-none-any.whl (2.7 kB)
Collecting excel-base>=1.0.3
  Downloading excel_base-1.0.4-py2.py3-none-any.whl (5.2 kB)
Collecting xlwt
  Downloading xlwt-1.3.0-py2.py3-none-any.whl (99 kB)
     ---------------------------------------- 100.0/100.0 kB 1.9 MB/s eta 0:00:00
Collecting screen
  Downloading screen-1.0.1.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Collecting TimeConvert>=1.5.3
  Downloading TimeConvert-3.0.13-py2.py3-none-any.whl (19 kB)
Collecting python-dateutil>=2.8.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ---------------------------------------- 247.7/247.7 kB 3.8 MB/s eta 0:00:00
Collecting tzlocal
  Downloading tzlocal-4.3-py3-none-any.whl (20 kB)
Collecting isoweek
  Downloading isoweek-1.3.3-py2.py3-none-any.whl (7.1 kB)
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting pytz-deprecation-shim
  Downloading pytz_deprecation_shim-0.1.0.post0-py2.py3-none-any.whl (15 kB)
Collecting tzdata
  Downloading tzdata-2022.7-py2.py3-none-any.whl (340 kB)
     ---------------------------------------- 340.1/340.1 kB 7.0 MB/s eta 0:00:00
Installing collected packages: xlwt, screen, isoweek, django-six, tzdata, six, pytz-deprecation-shim, python-dateutil, tzlocal, TimeConvert, excel-base, django-excel-response2, django-admin
  DEPRECATION: screen is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour cha
nge. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for screen ... error
  error: subprocess-exited-with-error

  × Running setup.py install for screen did not run successfully.
  | exit code: 1
  ╰─> [14 lines of output]
      running install
      C:\Users\Padmini Kuchukulla\AppData\Local\Programs\Python\Python310\lib\site-packages\setuptools\command\install.py:34: SetuptoolsDeprecationWarning: setup.py install is deprecated. Use build and pip and o
ther standards-based tools.
        warnings.warn(
      running build
      running build_py
```

After this installed the virtualwrapper, this is because we are creating our whole project in separate virtual environment using the following command

```
pip install virtualenvwrapper-win
```

After installing the wrapper for windows, now created an environment for our project using the following command

```
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)
     ---------------------------------------- 468.5/468.5 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)
  seeder 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,NushellActivator,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)
     ---------------------------------------- 42.8/42.8 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
```

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 1050, 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 1050, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 992, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
  File "<frozen importlib._bootstrap>", line 1050, 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\OnlineLibraryManagementSystem>pip install requests
Collecting requests
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
     ---------------------------------------- 62.8/62.8 kB 1.1 MB/s eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
     ---------------------------------------- 61.5/61.5 kB 3.4 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.1.0-cp310-cp310-win_amd64.whl (97 kB)
     ---------------------------------------- 97.1/97.1 kB 5.8 MB/s eta 0:00:00
Collecting certifi>=2017.4.17
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
     ---------------------------------------- 155.3/155.3 kB ? eta 0:00:00
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.15-py2.py3-none-any.whl (140 kB)
     ---------------------------------------- 140.9/140.9 kB 8.2 MB/s eta 0:00:00
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\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)
```

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\b773edc52acb16cd2db4cf1a0b47117e2f69bb4eb300ed0e70
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\Envs\penguins\lib\site-packages\django\db\backends\base\base.py", line 282, in ensure_connection
    self.connect()
  File "C:\Users\Padmini Kuchukulla\Envs\penguins\lib\site-packages\django\utils\asyncio.py", line 26, in inner
    return func(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\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\Envs\penguins\lib\site-packages\django\utils\asyncio.py", line 26, in inner
    return func(*args, **kwargs)
  File "C:\Users\Padmini Kuchukulla\Envs\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\Envs\penguins\lib\site-packages\psycopg2\__init__.py", line 122, in connect
    conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
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. Features and its Limitations:

- Voice Based Search: One of the most sophisticated feature of our project, was incredibly hard to implement because of API version mismatch. Might still have some issues in browsers that do not support the versions
- Key Based Search : Allows the user to search for books using the author namer or the isbn numbero the book.
- Students Access to Borrowed books: Gives the student a clear idea of the books they have loaned from the library along with the deadline they have to return the book
- Student Tracker: The admin can track the students borrowing books and notify them when they are close to the deadline.
- Payment Gateway : A feature we were working rigorously to implement for the fine to be collected, but unfortunately were not able to implement because of versions not matching for the payment API of the Razorpay open source software.
- Automatic Categorization: The books are categorized by different departments they belong to basically their genres and makes it easy for the user to implement.

g.Reflection:

In phase 2 we accomplished all the goals we had originally planned and further added the original phase 3 features to the already existing schedules and accomplished that too. In this phase we had worked in groups of 2 and 3 to accomplish the assigned tasks of web page integrations with the database which was created by the other groups and another team was working on the testing for phase 2.

While doing this, help could optimize the tables and web pages and all the errors that were faced with brainstorm meetings. In this collaborative manner we were capable of developing the whole original phase 2 with add ons from phase 3. We did add the alerts to already existing phase 1 front end code and generalized for the whole UIs.

The constant feedback and help from one another were able to complete the assigned tasks successfully with minimal redundancy and little code debt with improvement to the front end pages too. We believe that these changes will help to have a smooth implementation of phase 3 objectives and successful completion of the project as a whole.

Member's Contribution:

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Ashraf Syed	Requirements, Meeting minutes updation, involved in building the code and unit tests	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 from peer Evaluation, involved in building the code	14.3%	
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, Contributed in test cases.	14.3%	
Vamsi Venkat Manepalli	Involved in building the code, Contributed in test cases.	14.3%	
Sai Vishwak Jadala	Involved in building the code, Contributed in test cases.	14.3%	