CHAPTER 1

INTRODUCTION

Interaction among students and teachers plays a vital role in education. The NIE Hub is a one-stop online app to ease the communication not only among students but also among students and teachers of the National Institute of Engineering, Mysore. The NIE Hub can connect students among themselves, staff and faculty members.

The students face problems in accessing/getting updates about things happening in college. The purpose of this connection through the NIE Hub is manifold. Information on official notifications, transfer of notes from the faculty members, college fests/workshops, availability of resources, could easily and quickly be disseminated via the user-friendly NIE Hub. All this is being facilitated by letting the recognized user category to upload the information on the web app.

The NIE Hub keeps you posted about the latest college notifications, news, and updates. The app provides a universal platform for ease of interaction among students themselves, and also among the faculty members and students to pass on the information on class resources.

The NIE Hub app also facilitates the Event/Fest organizers to inform the concerned about the programs scheduled and sale of any item related to their fests, such as fest T-shirts and other similar products.

The highlight of the NIE Hub app is that it forms a bridge for communication between the teachers and the students. It provides a facility for the teachers to upload the resources for the target students, on to the web application. The students can view them online from where ever or whenever they want. This app saves the teachers from making their separate websites to upload such resources, saving their time and effort.

The web interface is built using HTML, CSS and JS. The databases are maintained by MYSQL and the website is hosted using the Django Framework for Python. Any fairly modern computer with an operating system that supports a web-browser can be used to access the website.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system of informing the students about the events going on in our college is carried out through posters put up on the notice board outside the Golden Jubilee Block. However, due to the busy schedules of the students and teachers they will not be able to see the notice board, hence missing out on the events. In addition to this, the merchandise of events such as shirts, caps, event passes are all sold near the entrance of Golden Jubilee Block. Since everyone is in a hurry to reach their classes in the morning and homes/hostel in the evening, generally students will not be aware and hence miss out on these things.

In addition, the existing system of teachers sharing reading material/resources with the students happen either through websites created by the teacher or by the teacher going all the way to the Photocopy shop and giving the material to share. Since there are a lot of teachers in NIE, each one creating a separate website and for each student to access different websites to obtain the required material would be time-consuming indeed.

The existing system of buying books for the new semester happens entirely offline, every student has to go to a book shop and buy the books for the semester. Seniors who can pass their books over to the juniors will not be able to do so because they are not well connected to the juniors in the current system.

Thus, it is construed that even though the present system is useful, this system is not very efficient as most of the tasks in this system are carried on offline. Therefore, having a single platform to carry out all the above mentioned tasks would be really efficient in terms of convenience, time and cost.

2.2 PROPOSED SYSTEM

This project aims at developing a user friendly software platform for easy access of information and material to the teachers and students. Dissemination of information through this modality is efficient in terms of time, effort and convenience.

The database keeps track of the various data required.

This project is divided into the following sections:

Login module: There is a page for the user to sign up for the application either as a teacher or as a student. This module authenticates the user to login to their respective profile using their login ids and password.

2.2.1. Posts/Information Section: In this section, teachers can upload study materials along with other information pertaining to students using this web interface denoting the semester and branch of the of the target students. These posts or announcements can be viewed by the students belonging to that semester and branch by logging into the website through their credentials.

2.2.2. Events Section: In this section, event organizers can post the events and the information about the events such as the information pertaining to the items being sold. After the event is posted, the users can then view the events going on and also book the merchandise sold by the event organizer. This can be sold offline, by mutual communication between the organizer and the buyer through phone call.

2.2.3. Books Section: In this section, students can post the books they want to sell quoting the price they want to sell it for and all the necessary details regarding the book. Other students who want to buy the book can search for the book in the books section and reserve them. The seller will then see the request for the book and the books can be sold offline through mutual communication between the seller and the buyer through a phone call.

2.3 SYSTEM REQUIREMENTS

The present project uses Python 3.6 for the backend. The Django library provides a hosting service for http and ftp responses. MySQL is used for the database. It is a RDBMS which can be easily integrated with Django Python. These two systems handle the server side of things. The frontend of our website uses HTML. Bootstrap is used to stylize the website. Custom CSS and JS components are also used where ever necessary.

The Django Project can be run on a server with network access. This is important because the website grabs CSS, JS and bootstrap files from third party servers. The website requires a fairly modern web browser with JS enabled. Details about the user's system requirements are discussed later in the report.

SOFTWARE DETAILS:

Program or Package	***Version Used***
Python	**3.6**
MySQL	**5**
Django	**2.0**
Bootstrap	**3.3.7**
HTML	**5**
CSS	**3**

,

CHAPTER 3

3.1 SYSTEM ARCHITECTURE AND EXPLANATION

The name of our database will be nie_hub. We store all the tables under this database.

TABLES USED IN DATABASE:

LOGIN SECTION

Field	Type	Null	Key	Default	Extra
uid	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(20)	NO		NULL	
last_name	varchar(20)	YES		NULL	
branch	varchar(3)	NO		NULL	
sem	int(11)	YES		NULL	
address	longtext	YES		NULL	
email	varchar(30)	NO		NULL	
phone_number	varchar(10)	NO		NULL	
usn	varchar(10)	NO		NULL	
password	varchar(20)	NO		NULL	
category	varchar(10)	NO		NULL	

Figure 3.1: User table describing the schema of the users.

The user table (Figure 3.1) stores all the information about the users who sign up for our website. It also provides necessary details to the buyers and sellers later when a transaction has to be made between them. This allows a user to login to the website through his credentials.

3.1.1. POSTS SECTION

Field	Type	Null	Key	Default	Extra
post_id	int(11)	NO	PRI	NULL	auto_increment
title	varchar(40)	NO		NULL	
date	datetime(6)	YES		NULL	
branch	varchar(3)	NO		NULL	
sem	int(11)	NO		NULL	
user_id_id	int(11)	NO	MUL	NULL	
body	longtext	YES		NULL	

Figure 3.2: Posts table describing the schema of uploaded posts

The Posts table (Figure 3.2) stores all the information about the posts created by the teachers. It stores the title of the post, body of the post, date of creation of the post, branch and semester for which the post is targeted along with the user_id which is the id of the user creating the post. Hence, the user id field should be a foreign key to the user table.

Field	Type	Null	Key	Default	Extra
attachment_id	int(11)	NO	PRI	NULL	auto_increment
post_id_id	int(11)	NO	MUL	NULL	
attachment_link	varchar(100)	YES		NULL	

Figure 3.3: Attachments table describing the schema of uploaded attachments

The Attachments table (Figure 3.3) is needed to store the urls of the files that are uploaded as attachments. The urls store the address of the file location in the server. The post_id is a foreign key to posts table denoting the post to which the attachment belongs to.

3.1.2. EVENTS SECTION.

Field	Type	Null	Key	Default	Extra
event_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(40)	NO		NULL	
body	longtext	YES		NULL	
event_date	datetime(6)	YES		NULL	
create_date	datetime(6)	YES		NULL	
venue	varchar(30)	YES		NULL	
interested_count	int(11)	NO		NULL	
owner_id_id	int(11)	NO	MUL	NULL	

Figure 3.4: Events table describing the schema of uploaded events

The events table stores the necessary information about the created events. It maps the owner_id to the user table denoting the creator of the event.

Field	Type	Null	Key	Default	Extra
item_id	int(11)	NO	PRI	NULL	auto_increment
category	varchar(30)	NO		NULL	
name	varchar(30)	YES		NULL	
event_id_id	int(11)	NO	MUL	NULL	

Figure 3.5: Items table describing the schema of uploaded items

The Items Table contains only the name and category of the item and has a foreign key event_id mapping an item to its respective event.

Field	Type	Null	Key	Default	Extra
item_details_id	int(11)	NO	PRI	NULL	auto_increment
size	varchar(3)	YES		NULL	
quantity	int(11)	NO		NULL	
price	int(11)	NO		NULL	
item_id_id	int(11)	NO	MUL	NULL	

Figure 3.6 Item_details table describing the schema of items in detail

The item_details table stores the item specific information such as size, quantity and price. It has a foreign key item_id mapping the details to the item.

Field	Type	Null	Key	Default	Extra
item_waiting_id	int(11)	NO	PRI	NULL	auto_increment
request_quantity	int(11)	NO		NULL	
date	datetime(6)	YES		NULL	
buyer_id_id	int(11)	NO	MUL	NULL	
item_details_id_id	int(11)	NO	MUL	NULL	

Figure 3.7: Waiting_items table describing the schema of booked items yet to be issued by seller

The waiting_items table contains all the items that are booked by the buyer but payment and selling of the item is not yet confirmed by the customer. It contains the details of the booked items.

Field	Type	Null	Key	Default	Extra
item_trans_id	int(11)	NO	PRI	NULL	auto_increment
date	datetime(6)	YES		NULL	
quantity_sold	int(11)	NO		NULL	
size	varchar(3)	YES		NULL	
price	int(11)	NO		NULL	
category	varchar(30)	NO		NULL	
name	varchar(30)	YES		NULL	
buyer_id_id	int(11)	NO	MUL	NULL	
owner_id_id	int(11)	NO	MUL	NULL	

Figure 3.8: Item_transactions table describing the schema of transactions of items

This Item_transanction table contains details about the Items that have been confirmed to be bought by the buyer.

3.1.3. BOOKS SECTION

Field	Type	Null	Key	Default	Extra
book_id	int(11)	NO	PRI	NULL	auto_increment
branch	varchar(3)	NO		NULL	
sem	int(11)	NO		NULL	
title	varchar(50)	NO		NULL	
author	varchar(30)	NO		NULL	

Figure 3.9: Books table describing the schema of uploaded books

The Books table stores the necessary information about the books.

Field	Type	Null	Key	Default	Extra
book_details_id	int(11)	NO	PRI	NULL	auto_increment
edition	int(11)	NO		NULL	
price	int(11)	NO		NULL	
status	int(11)	NO		NULL	
book_id_id	int(11)	NO	MUL	NULL	
owner_id_id	int(11)	NO	MUL	NULL	

Figure 3.10: Book_details table describing the schema of books in detail

This Books_details table stores the information specific to a book. The status denotes if the book is available for sale (1) or if it has been booked by someone else.

Field	Type	Null	Key	Default	Extra
book_trans_id	int(11)	NO	PRI	NULL	auto_increment
date	datetime(6)	YES		NULL	
title1	varchar(60)	YES		NULL	
author1	varchar(30)	YES		NULL	
edition1	int(11)	YES		NULL	
price1	int(11)	YES		NULL	
buyer_id1_id	int(11)	NO	MUL	NULL	
owner_id1_id	int(11)	NO	MUL	NULL	

Figure 3.11 Transaction_books table describing the schema of transactions of books

The Transanction_books table contains details about the Books that have been confirmed to be bought by the buyer.

Field	Type	Null	Key	Default	Extra
book_wait_id	int(11)	NO	PRI	NULL	auto_increment
date	datetime(6)	YES		NULL	
book_details_id_id	int(11)	NO	MUL	NULL	
buyer_id_id	int(11)	NO	MUL	NULL	

Figure 3.12: Waiting_books table describing the schema of booked books yet to be issued by seller.

The Waiting_books table contains all the items that are booked by the buyer but payment and selling of the book is not yet confirmed by the customer. It contains the details of the booked items.

3.2 Entity Relationship (ER) Diagrams

3.2.1. USER AND POSTS:

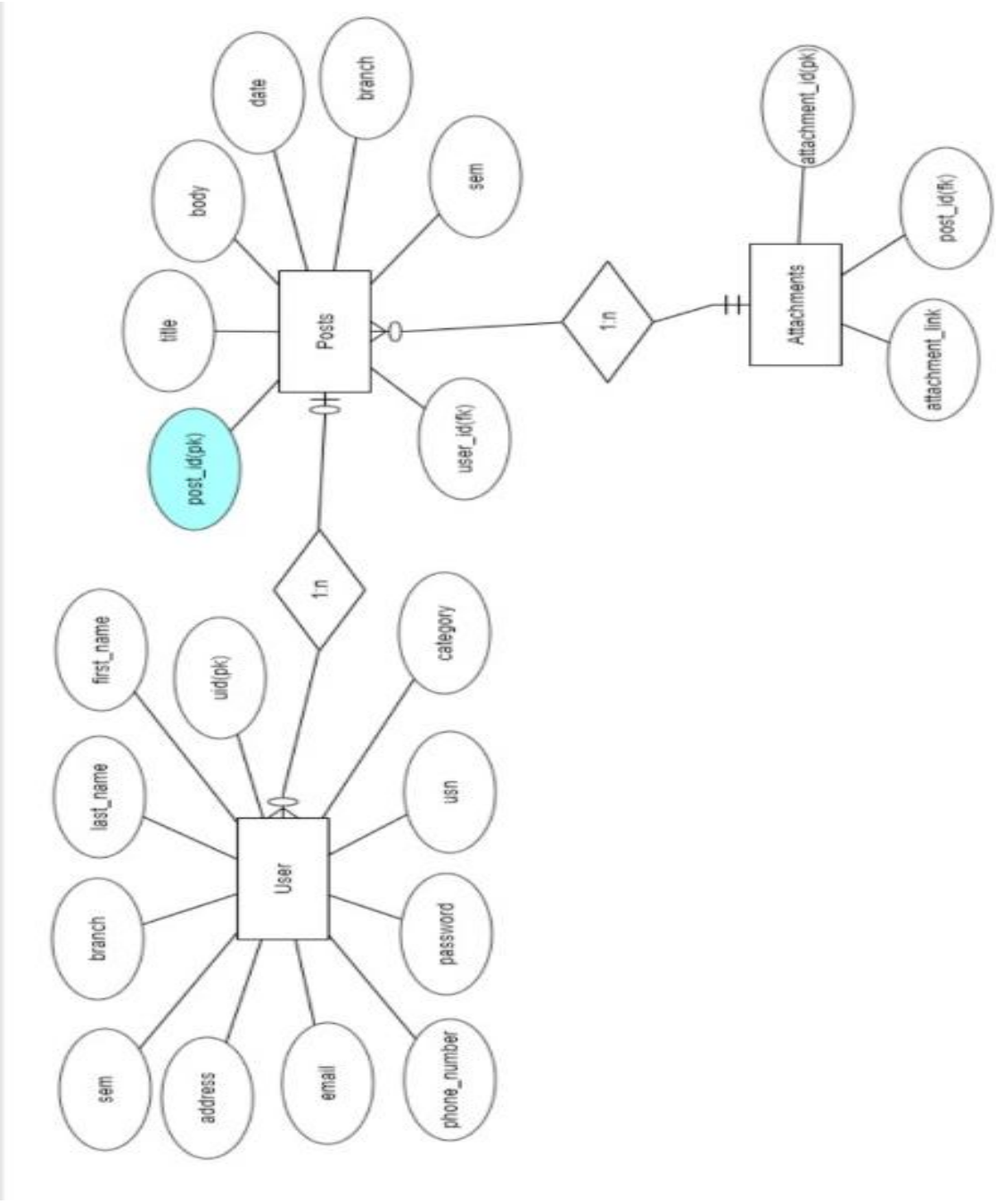

Figure 3.2.1: ER diagram of user and posts

3.2.2. EVENTS:

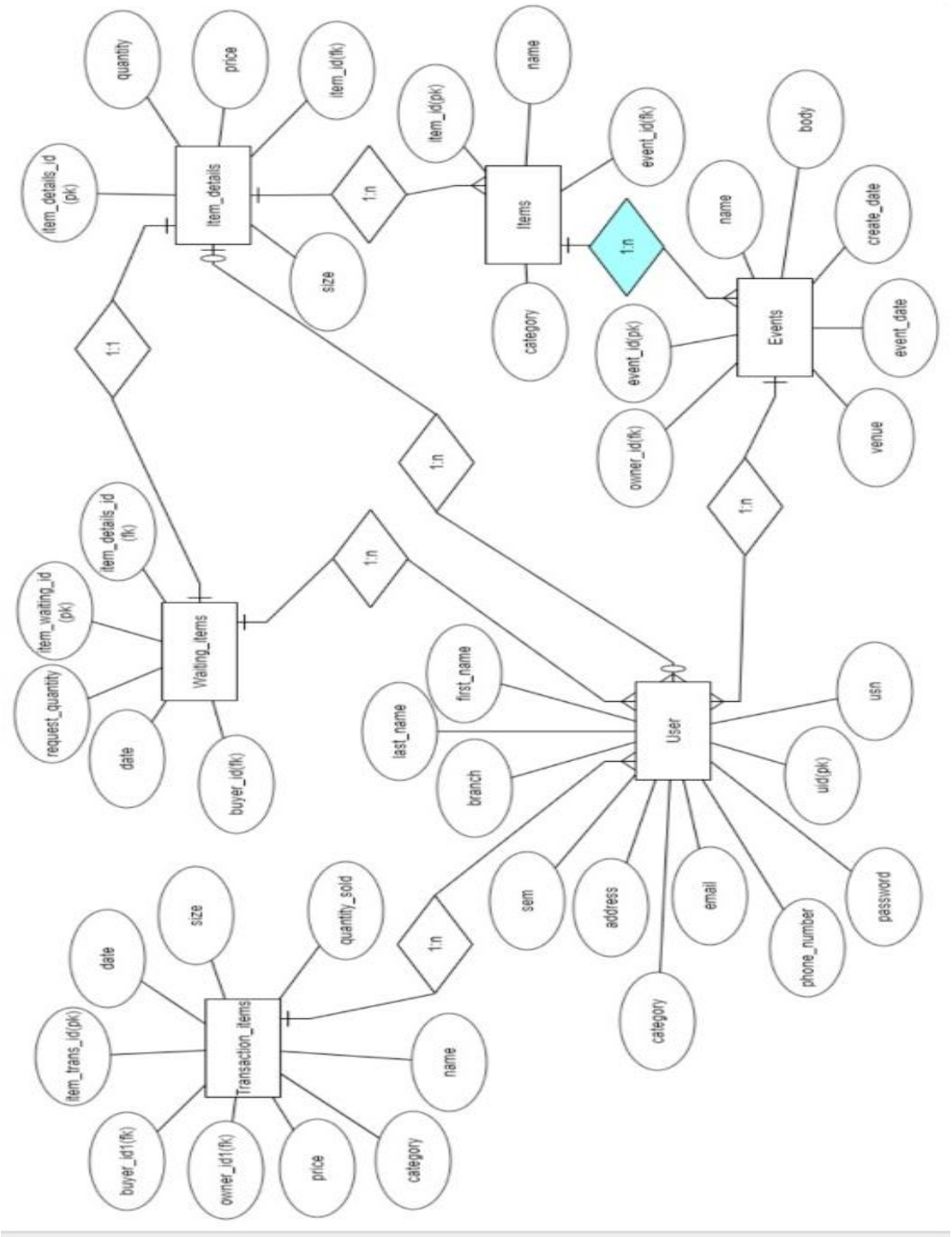

Figure 3.2.2: ER diagram of events

3.2.3. BOOKS:

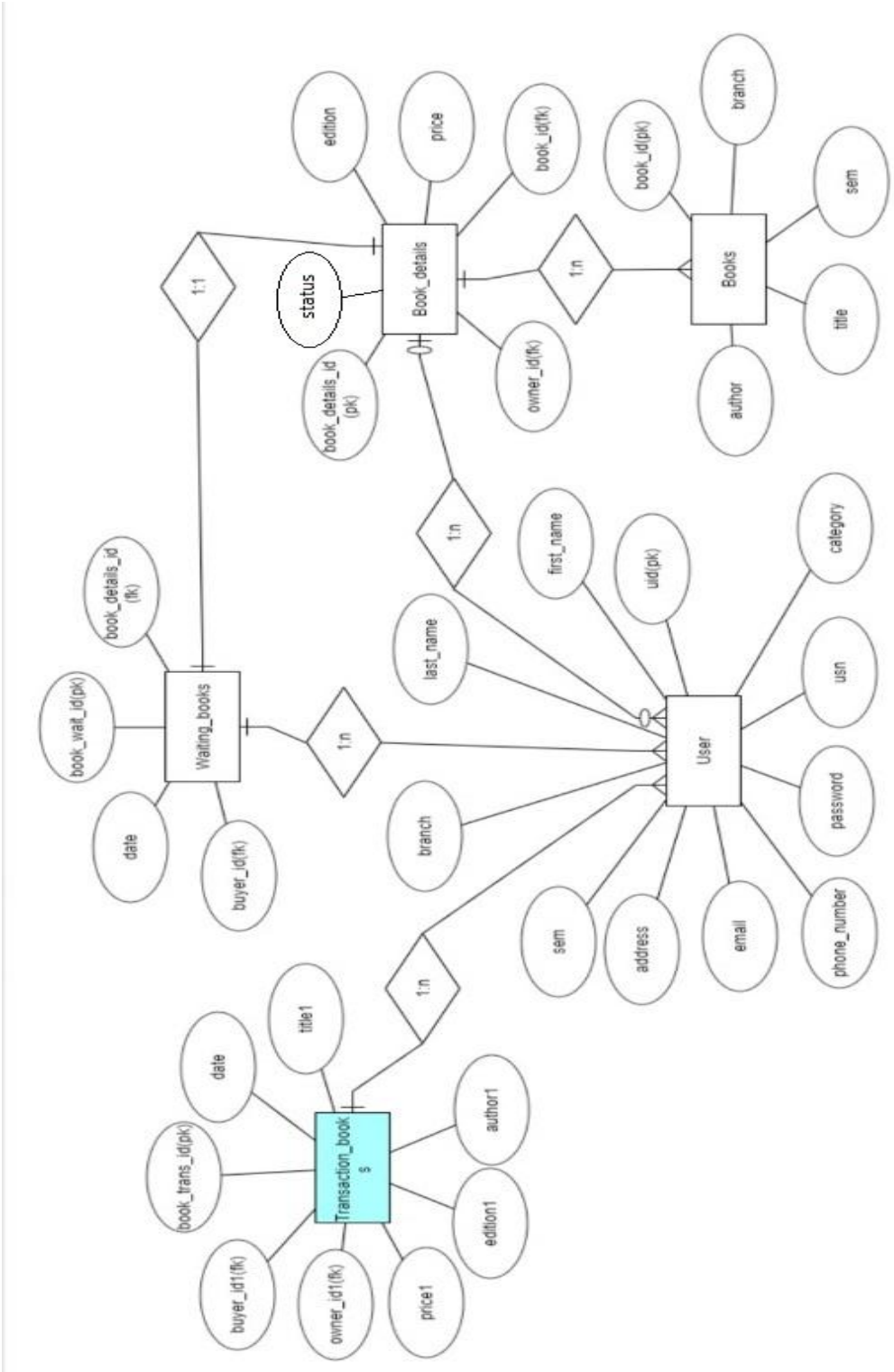

Figure 3.2.3: ER diagram of books

CHAPTER 4

SYSTEM IMPLEMENTATION

The present project uses Python with Django Framework for the backend. Django is a free and open source web application framework, written in Python. A web framework is a set of components that helps to develop websites faster and easier. Django is flexible and allows building powerful websites. It can generate dynamic content, collect form data and modify the database effectively using various libraries. The software used includes Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and Bootstrap for the front end. For managing the database, the MySQL is being used. CSS components are used for styling the website. The project runs on a local server.

Django

Django is a Python-based free and open-source web framework, which follows the model-view-template (MVT) architectural pattern.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Before we install these pip packages we need to make sure our system is setup to connect to MySQL, we need to ensure we have the correct MySQL Python connector. After installing the correct version of MySQL, we need to execute this command.

pip install django mysqlclient

This will download and install the packages. We will now set-up our test application and store the files in the app folder.

django-admin.py startproject project_name

To check it is working we will simply make migrations and migrate

python manage.py makemigrations

python manage.py migrate

MYSQL

Relational database InnoDB storage engine is used because of its fast performance because InnoDB arranges our data on disk to optimize common queries based on primary keys Each InnoDB table has a primary key index called the clustered index that organizes the data to minimize I/O for primary key lookups and rich feature set like it provides support for FOREIGN KEY referential-integrity constraints also its design follows the ACID model, with transactions featuring commit, rollback, and crash-recovery capabilities to protect user data. MYSQL is used for database functionality.

HTML5

HTML or HYPERTEXT MARKUP LANGUAGE is the standard mark-up language used to create webpages. The purpose of the web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, mailing it a markup language rather than a programming language.

Cascading style sheets

Cascading style sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interface for web applications, and user interface for many mobile applications. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics ,enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. **It** is a free and open-source front-end framework for designing websites and web applications. It

contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only.

CHAPTER 5

TESTING

Test Cases:

The website was tested with a set or random generated data. The data included a multitude of cases to ensure that all things run as expected. Test cases of leaving the website without logging out or trying to extract user data without a proper login id and/or password are also tested.

The registration forms were also tested with improper entries like entering a string of characters instead of the phone number. If there is any discrepancy in the entered data, the entered form is not processed and a proper error message is sent back to the user.

In the posts section, it is made sure that the users who haven't logged in as teachers will not be able to upload posts by any means. Similarly, teachers will not be allowed to view posts by other teachers. If no posts were found a message is displayed on the page saying no posts are found.

The attachments uploaded for each post is made sure to be displayed when the post is viewed.

In the events section, all the events with a date further than the present date are made sure to stay on the top as the past events will be of very less importance. Also, if no events are found a user friendly message saying no events found is displayed so that the user does not get lost while using the website. It is made sure that the events created by user are not displayed for him as it may confuse the user.

In addition, when entering details for an item, if a user wishes to update a related item of different sizes for example shirt, he is allowed to do so without any discrepancies.

The book selling section was tested for adding new books by the users. The system was tested so as to check its integrity when displaying the valid books or returning a

relevant error message if no such book was found to the user. The book is added back to the database when the seller decides he does not want sell his books by pressing the abort button.

It is made sure that a book which is sold by the user will not be listed in the set of books which the same user wants to buy.

Futher, the buy book section was checked to allow only those books which have not been sold to anybody to be removed. The buy book functionality was tested to check to allow only those books which were not booked previously by other users.

In the profile section, the user who wants to change password can change it by clicking on the “Change Password” button. His current password is validated and if it is valid “new password” and “confirm password” inputs are checked. If any of these entries are not valid, user is displayed with an error message and is redirected to the “Change Password” page again.

CHAPTER 6

RESULT

User:

Sign up Section:

The User has to first Sign in to create an account in The NIE HUB app.

Login Section:

Once the user has created an account, he/she is ready to login.

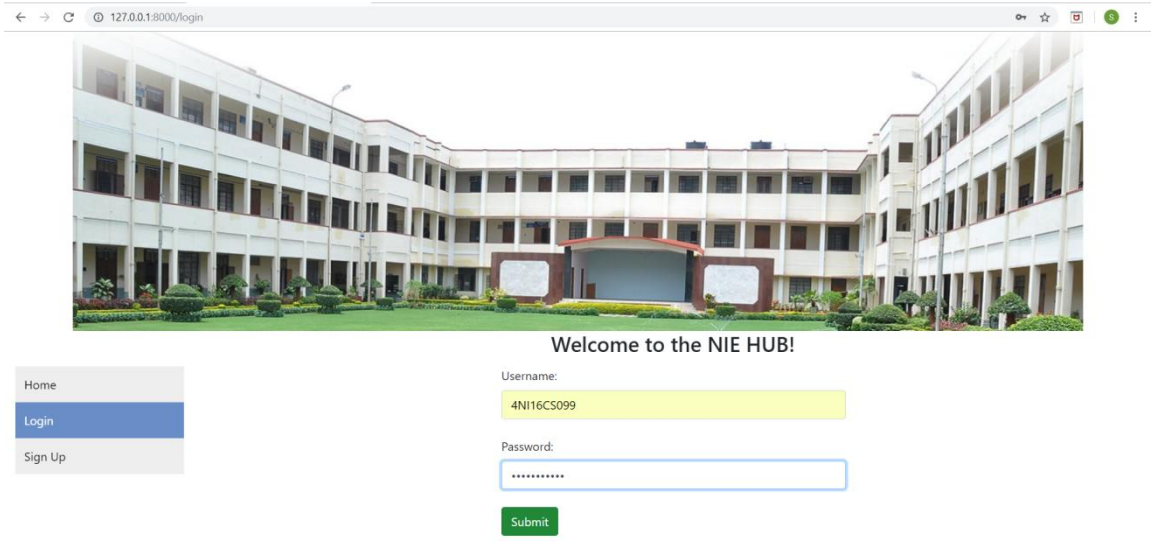

Figure 6.1: Screen shot of the login page.

Posts:

In this section, a student can view if he/she has received any notes or lecture material/ resource or any important information/ announcement from his/her teacher.

Create Post: This option is available only to the teachers of NIE to share their notes and lecture material. This can be used to send notes only to a particular semester and branch as an option is provided in the app to send only to target students belonging to a particular branch and semester.

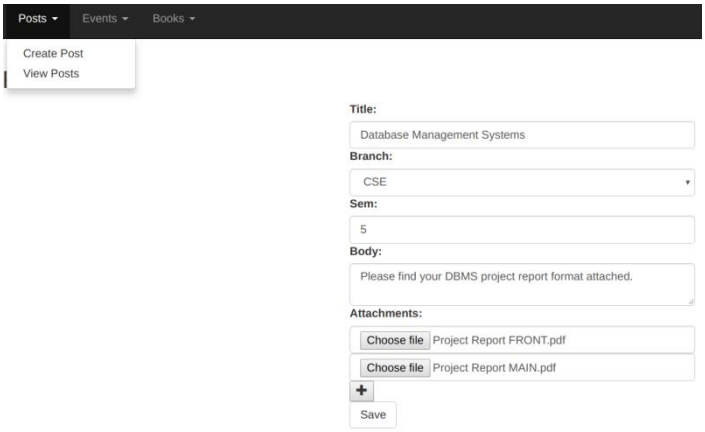

Figure 6.2: Screen shot of creating posts.

View Post: Using this option, the students can view posts/notes received from their teachers.

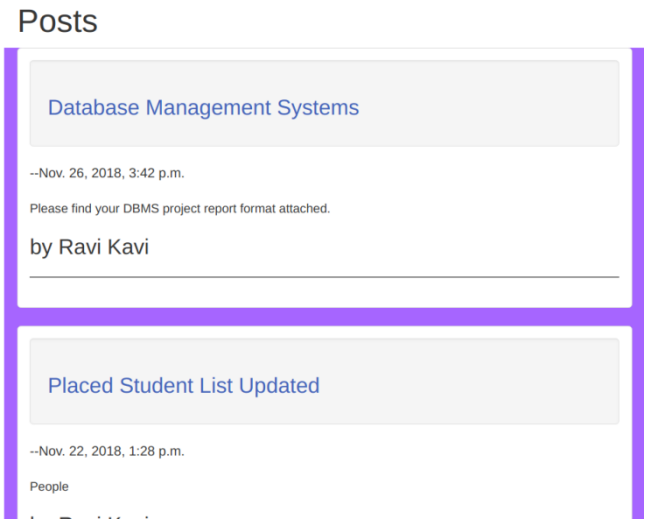

Figure 6.3: Screen shot of viewing posts

Books:

In this section, the user can sell and buy books (if he does not know what to buy, he/she can view and select the books based on the specific semester; or if he/she knows what book he/she wants to buy, he/she can enter the details of the book).

Sell Books: The user selling the book has to enter the book details that is, Book title, author, year of publication, edition, price and to which branch and semester it belongs to.

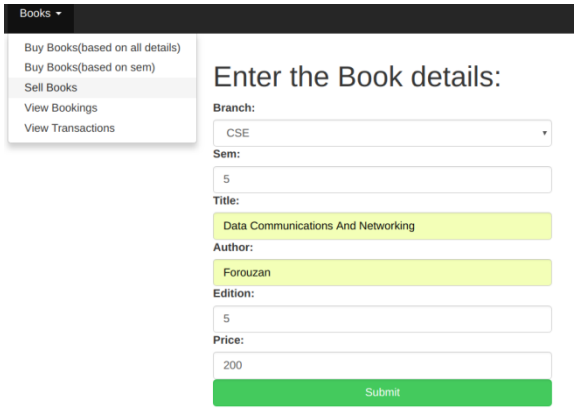

Figure 6.4: Screen shot of entering the book details for selling.

Buy Books (based on semester): The user who is unaware of what books to buy can just enter the semester he is in so that he can view all books in the buying section belonging to that semester.

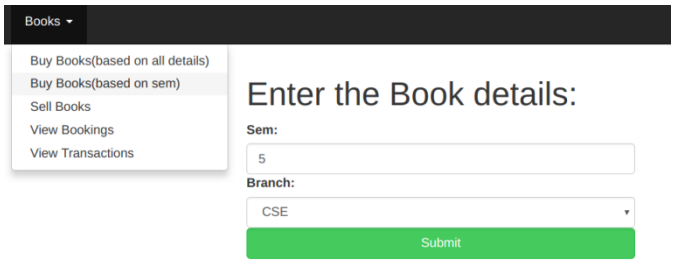

Figure 6.5: Screen shot of buying books based on specific semester.

Buy Books (based on all details): The user who is sure of what book to buy can enter the book name, author, year and edition and view book details in the buying section matching these conditions. If the user finds the book price to be reasonable and acceptable, then he can book it by clicking the “Yes” Button in the “Book it” column.

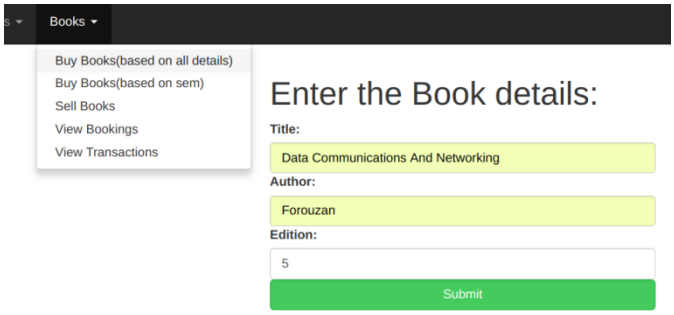

Figure 6.6: Screen shot of buying books based on details of the books.

View Bookings: Once the buyer has agreed to buy it, the seller is notified and he can view in the View Bookings section as to who the seller is and which book of his has been booked by the buyer. For convenience, both the seller and buyer are provided with each others' mobile numbers rather than giving it only to the seller, so that they can decide where to meet, collect and make the transaction.

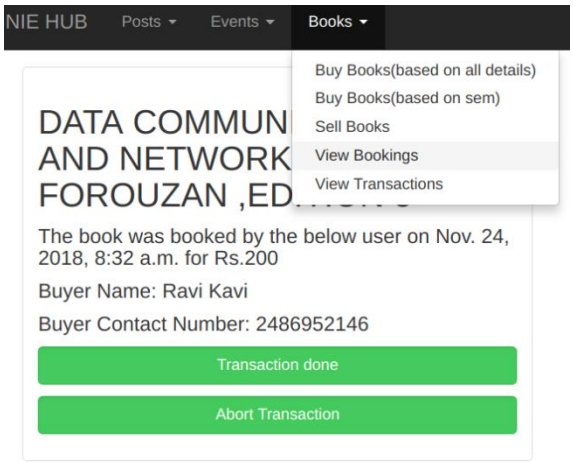

Figure 6.7: Screenshot for viewing the bookings.

View Transactions: Once they have made the transaction, the seller has to click the "Transaction Done" button in the View Bookings section to confirm the transaction and all these details will be inserted into the Transaction_Books table. In case the seller is not ready to sell the book he can click on the "Abort Transaction" button in the View Bookings section and there will not be any transaction happening.

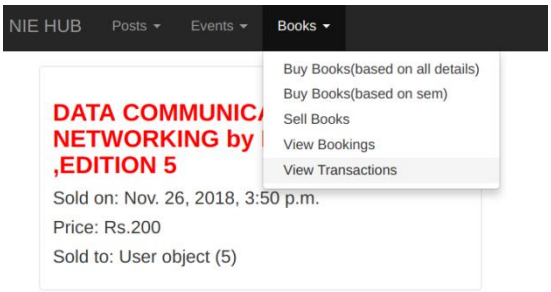

Figure 6.8: Screen shot of viewing transactions.

Events:

In this section, people can get to know about the latest events, workshops, competitions, sport events and other college activities/events. Through this app, the people can now be aware of all the events that are happening in the college and not miss out on the fun.

Create Event: A user (Event Manager) can create an event by entering the name, event date, venue and details of the event. Information about the event is made available to all by making this event information to be in the "View Events" section of every account. The Event Manager can also sell items related to the event so people interested in buying those items can pre-book it.

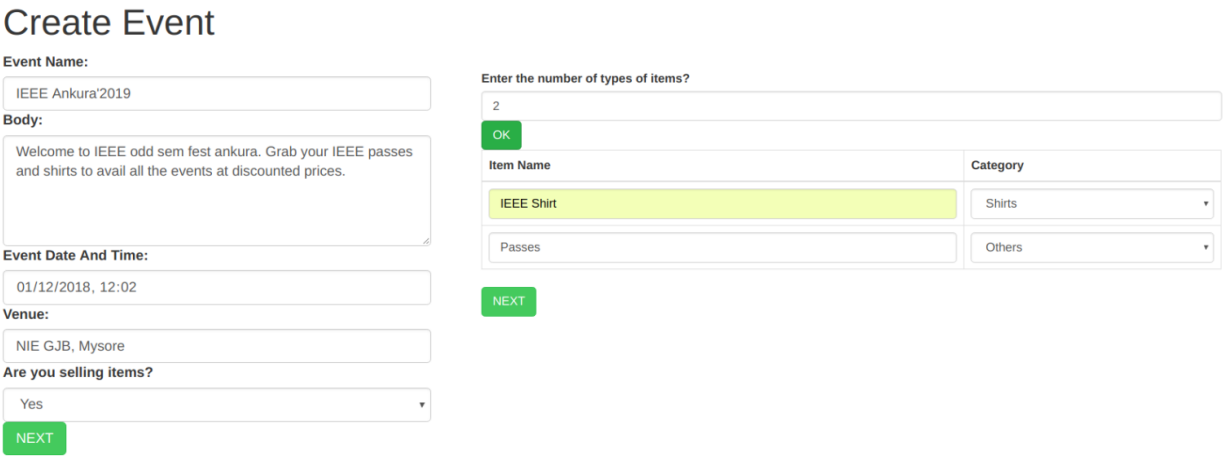

Figure 6.9a: Screen shot of creating an event and its details.

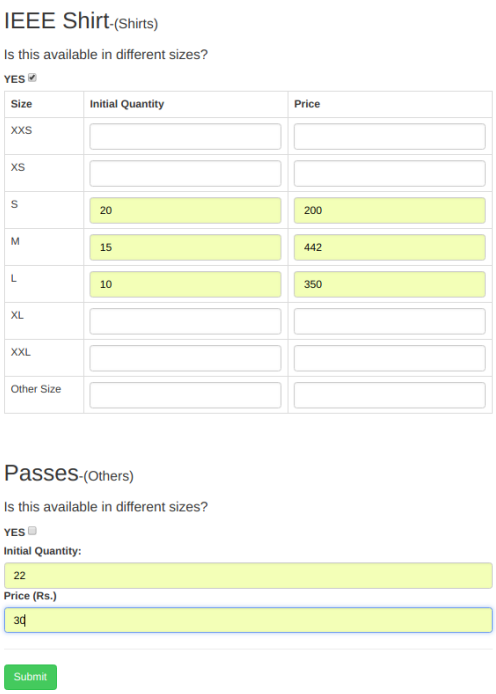

Figure 6.9b: Screen shot of creating an event and its details.

View Events: Here, the user can view all the events that are created by other users. "View events" is segregated based on Upcoming (Events which have not yet occurred) and Past (Events which have occurred/finished) events. On clicking the event, a detailed view of the events is displayed where he/she can book the items.

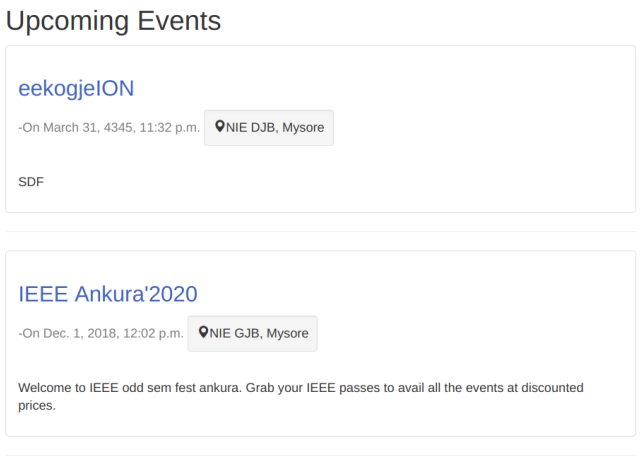

Figure 6.10: Screen shot of details of upcoming events.

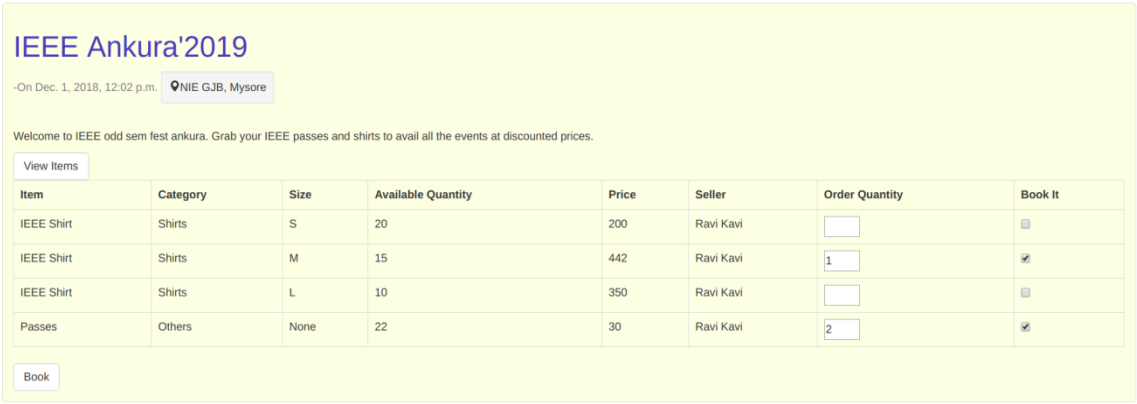

Figure 6.11: Screen shot of displaying merchandise related to an event available for booking.

Manage Event: The Event Manager can keep track of how many items are still available in stock. And Issue the requested number of items by pressing the issue button.

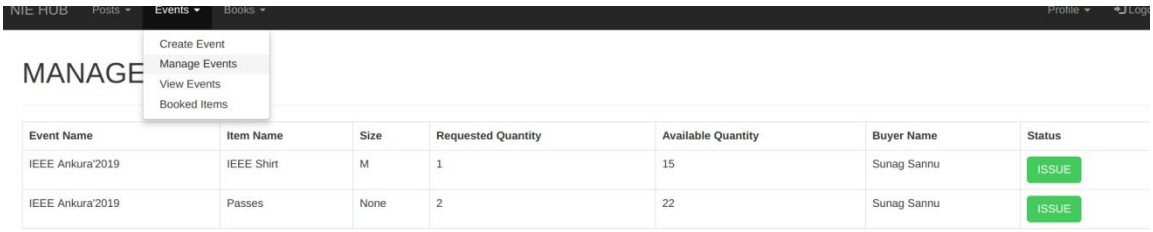

Figure 6.12: Screen shot of managing events.

Booked Items: Once the people have booked the items and have paid the amount to the event organizers, the organizers have to ISSUE the items. Once a transaction is completed, all the transactions appear under booked items.

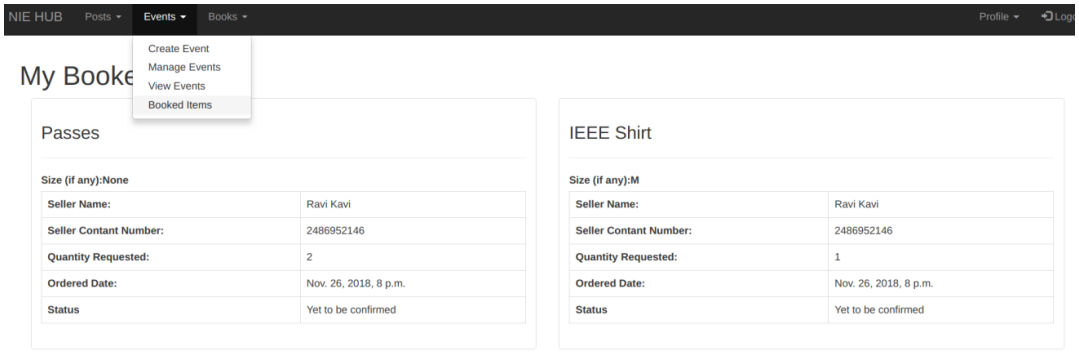

Figure 6.13: Screen shot of booked items.

Profile:

User can view his bookings, his profile and the events he has hosted in this section of our app.

In the My Books, My Events and My Posts (only for a teacher) sections user is allowed to delete his book put for sale, remove an event hosted and delete a post sent to students respectively if he finds its necessary. These updates are made in the database by removing such entries.

Figure 6.14: Screen shot for updating the book status and the events.

CONCLUSION AND FUTURE ENHANCEMENTS

The purpose of the website, titled the NIE Hub, was to create a one stop application mainly to improve the connections between the students and teachers and make their life simpler by improving the way of sharing information and resources.

This application can be enhanced by providing features like chat platform where students may chat among themselves and lend their merchandise/ books. This would reduce the offline dependency of the app. We could also incorporate logistics to facilitate the transport of the merchandise/ books to the address of the buyer.

A good future improvement would be to include a payment method directly in the website. This will increase the usability of the website and will save a lot of time for the users. Incorporating this feature requires in the application requires a lot of time and money to set up and hence could not be done within the short time we had.

REFERENCES

Narain Gehani (2008). The Database book: Principles and Practice using MySQL, Universities Press (India) Private Limited.

Elmasri R and Navathe SB (2007). Fundamentals of Database System, 5th edition. Boston: Addison-Wesley.

https://www.w3schools.com/js/ for JavaScript tutorials.

https://getbootstrap.com/docs/3.3/components/ for bootstrap documentation.

https://www.youtube.com/watch?v=FNQxxpM1yOs&vl=en Sentdex channel for Django tutorials.

https://tutorial.djangogirls.org/en/ for Django examples.

https://docs.djangoproject.com/en/2.1/ for Django documentation.

https://stackoverflow.com/ to solve common problems.