Chapter 1: INTRODUCTION

1.1 Project Overview

The Project “RIMT University Dashboard” Is Basically Made for a new Module of Current ERP of our University which make possible to provide a system with a good user interface for user to interact with the academic’s system to view their data regarding University, like Attendance, Projects, Assignment, Hostel Facility, Bus facility, and these all will have integrated with a new module called University Dashboard where each and every user of University

Dashboard user able to create a specific account for them and after that they will redirected to login page where they will login to their account and after that the user redirected to the Core ERP home page and there he/she can view their academics details like attendance, Course he/she register for, Pending project or assignment if any, upcoming event in university or any holidays. And in the centre of the home page you will find a Real Time News feed where they can post their own pictures and Regarding their status and after there every member of the university can view that posts and Like, Comments on that post and Many More.

1.2 Purpose

The main purpose to make this project is make a separate news feed for college life regarding their academic’s life inside the RIMT University and integrated with real time data of each and every member of this origination. Because today’s world in each social media there are so many fake user account and useless data are available for our daily life news feed but if our project is connected with our ERP system then only a limited number of people which is already member of university is able to create their real account share only valid and useful data.

1.3 Scope

The Scope of this project is that it could be used in Our University ERP integrated with it and make our ERP system more interesting for everyone to use it and share useful information with all the member of origination.

It could be used in any origination as well and work with that origination internal member and also user is able to share their idea and according to which the other member of origination can show interest and comments there view on that idea.

This project would be very useful for educational institutes where regular evaluation of students' is required. Further it can also be useful for anyone who requires feedback based on objective type responses.

1.4 Definitions, Acronyms, And Abbreviations

The subsection provides the definitions of all terms, acronyms, and abbreviations used in this document to understand the SRS properly.

Sr. No.	Terms/Acronyms	Description
1	Student	User mostly a student who will appear for the University Member
2	Faculty	Another user mostly faculty member, lecturer or other person of origination.
3	Administrator	Super user, adds faculty and manages system.
4	Interface	Browser View where end user able to view output data like attendance, feeds etc.
5	Backend	Where the actual software run on to provide out to the user
6	Database	It is used for storing the user data regarding all information of the origination
7	Browser	User end Software where user able to view output.
8	Server	Software at the end of machine where actual program will run to provide output to the user.

Chapter 2 OUR OBJECTIVE

2.1 Objective

In this project its totally depend on the university or any specific origination and to use this software it provides the main and most basic task to the end user as service to help them for easily access to the information regarding their organization. Some of main object of "University Dashboard are following: -

2.2 Create new user:

Initially user enter this website/web application they will find a signup page where they will inter their details like Name, Email ID, Username, Password etc to set this data for further uses in the next login and feed view parts.

2.3 User login

After Creating new user. User will have redirected to the login page and their system ask them to Enter their username and password of his/her account which is set by the use at the time of creating their account.

2.4 News feed view

When user will successfully Login to the system user will redirected to the news feed pages where they will able to create new post by just clicking on create new post, or they will view different types of post posted by other member of same origination in the centre of the page.

If user will look at the right side at top it will show the upcoming event in origination with a photograph with details and date of event. and in the bottom right of the page user can view their pending assignment and project and also can view and download the Class note and E-books provide by the University.

And in left Sidebar at top section shows their user own details like name, profile picture, email address, username, current time and date. And it bottoms part user can view their academics attendance, course there which he or she register and also can able to view some top trending fest rezoned in past or in future by their origination.

2.5 Post new status

If user click on post new status button on the top of the news feed page they will redirected to a new page where they need to provide the details location of image which user want to share and also they can enter relevant text to images and click to post now button and after a second that post are shared will all the people inside the university/ origination.

2.6 Like & comments

User also able to like any post of other user and also comment if she or he want to say something regarding that post to the other user or user can share their view regarding that post.

Chapter 3 TECHNOLOGY REQUIRED

3.1 Hardware Requirement: –

- 20 Gb Including the Space Required to. Store the Data to The Database.
- RAM Minimum 4GB
- Server
- Valid Internet connection.

3.2 Software Requirement: –

- Operating System Minimum Windows 7
- PyCharm IDE
- SQLite3, MySQL, PostgreSQL Database
- Odoo Server.
- Anaconda Library

3.3 Language/Skills Requirement: –

- Python
- Django
- HTML
- CSS
- JavaScript
- Odoo

3.4 Browser Compatibility

The project being web based required compatibility with at least the popular web browsers. Microsoft Windows XP and above, Linux and Macintosh being the current popular operating system and Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari and Google Chrome being the currently popular web browsers.

Operating System → / Browsers ↓	Win 2000	WinXP	WinXPSP2	Win Vista	Win 7	Mac OS	Linux
	Modern Browsers						
IE 8.0	N/A	SUPP	SUPP	SUPP	SUPP	N/A	N/A
IE 7.0	N/A	N/A	N/A	N/A		N/A	
IE 6.0	N/A	N/A	N/A	N/A		N/A	
Firefox 3.5	N/A	SUPP	N/A	N/A		N/A	
Opera 9.23	N/A	SUPP	N/A	N/A		N/A	
Safari 9.27	N/A	SUPP	N/A	N/A		SUPP	
	"Legacy" Old Browsers						
IE5.5	N/A	N/A	N/A	N/A		N/A	
Netscape	N/A	N/A	N/A	N/A		N/A	

Chapter 4: IMPLEMENTATION

4.1 Software Development Life Cycle

The Systems Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved including the waterfall model (the original SDLC method), rapid application development(RAD), joint application development (JAD), the fountain model and the spiral model. Mostly, several models are combined into some sort of hybrid methodology.

Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely particular plan was followed. The following figure shows a general life cycle Process in software development:

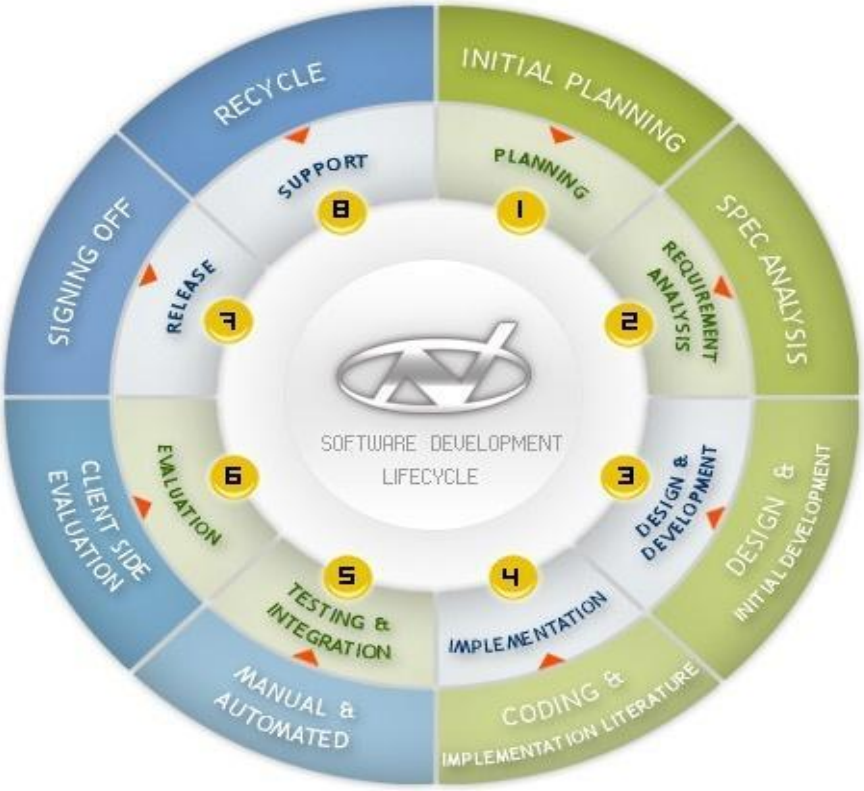


Figure 4.1-Software Development Lifecycle

The most common steps in all the development methodologies are as follows:

4.2 System/Information Engineering and Modelling:

As software is always of a large system (or business), work begins by establishing the requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when the software must interface with other elements such as hardware, people and other resources. System is the basic and very critical requirement for the existence of software in any entity. So if the system is not in place, the system should be engineered and put in place. In some cases, to extract the maximum output, the system should be re-engineered and spruced up. Once the ideal system is engineered or tuned, the development team studies the software requirement for the system.

4.3 Software Requirement Analysis:

This process is also known as feasibility study. In this phase, the development team visits the customer and studies their system. They investigate the need for possible software automation in the given system. By the end of the feasibility study, the team furnishes a document that holds the different specific recommendations for the candidate system. To understand the nature of the program(s) to be built, the system engineer or "Analyst" must understand the information domain for the software, as well as required function, behaviour, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved.

4.4 System Analysis and Design:

In this phase, the software development process, the software's overall structure and its nuances are defined. A software development model is thus created. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

Chapter 5: CODE GENERATION

The design must be translated into a machine-readable form. The code generation step performs this task. If the design is performed in a detailed manner, code generation can be accomplished without much complication. Programming tools like compilers, interpreters, debuggers etc are used to generate the code. Different high-level programming languages are used for coding. With respect to the type of application, the right programming language is chosen.

5.1 Introduction to Python

5.1.1 Introduction

Python is a simple, easy to learn, powerful, high level and object-oriented programming language. Python is an interpreted scripting language also. Guido Van Rossum is known as the founder of python programming.

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language. Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. Python’s syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas. Python supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles.

Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to declare an integer value in a variable. Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

5.1.2 Python Features

There are a lot of features provided by python programming language.

1) **Easy to Use:** Python is easy to very easy to use and high-level language. Thus it is programmer-friendly language.
2) **Expressive Language:** Python language is more expressive. The sense of expressive is the code is easily understandable.
3) **Interpreted Language:** Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
4) **Cross-platform language:** Python can run equally on different platforms such as Windows, Linux, Unix, Macintosh etc. Thus, Python is a portable language.
5) **Free and Open Source:** Python language is freely available (www.python.org).The source-code is also available. Therefore, it is open source.
6) **Object-Oriented language**: Python supports object oriented language. Concept of classes and objects comes into existence.
7) **Extensible:** It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in your python code.
8) **Large Standard Library:** Python has a large and broad library.
9) **GUI Programming:** Graphical user interfaces can be developed using Python.
10) **Integrated:** It can be easily integrated with languages like C, C++, JAVA etc.

5.1.3 Python OOPs Concepts

Python is an object-oriented programming language. You can easily create and use classes and objects in Python.

Major principles of object-oriented programming system are given below:

1) Object
2) Class
3) Method
4) Inheritance
5) Polymorphism

6) Data Abstraction

7) Encapsulation

1) Object

Object is an entity that has state and behaviour. It may be anything. It may be physical and logical. For example: mouse, keyboard, chair, table, pen etc.

Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute __doc__, which returns the doc string defined in the function source code.

2) Class

Class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class then it should contain an attribute and method i.e. an email id, name, age, salary etc.

3) Syntax:

```
class Class Name:
    <statement-1>
    .
    .
    <statement-N>
```

4) Method

Method is a function that is associated with an object. In Python, method is not unique to class instances. Any object type can have methods.

5) Inheritance

Inheritance is a feature of object-oriented programming. It specifies that one object acquires all the properties and behaviours of parent object. By using inheritance, you can define a new class with a little or no changes to the existing class. The new class is known as derived class or child class and from which it inherits the properties is called base class or parent class. It provides reusability of the code.

6) Polymorphism

Polymorphism is made by two words "poly" and "morphs". Poly means many and Morphs means form, shape. It defines that one task can be performed in different ways. For example: You have a class animal and all animals talk. But they talk differently. Here, the "talk" behaviour is polymorphic in the sense and totally depends on the

animal. So, the abstract "animal" concept does not actually "talk", but specific animals (like dogs and cats) have a concrete implementation of the action "talk".

7) **Encapsulation**

Encapsulation is also the feature of object-oriented programming. It is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.

8) **Data Abstraction**

Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonym because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things, so that the name captures the core of what a function or a whole program does.

5.2 Introduction to Django

5.2.1 Django framework

Django is an open source web application frame work written in Python. The primary goal of Django is to make the development of complex, data-based websites easier. Thus, Django emphasizes the reusability and pluggability of components to ensure rapid developments. Django consists of three major parts: model, view and template

5.2.2 Model

Model is a single, definitive data source which contains the essential field and behaviour of the data. Usually one model is one table in the database. Each attribute in the model represents a field of a table in the database. Django provides a set of automatically-generated database application programming interfaces (APIs) for the convenience of users.

```
# Signup UserModel Here
class UserModel(models.Model):
    name = models.CharField(max_length=150)
    username = models.CharField(max_length=150)
    email = models.EmailField()
    password = models.CharField(max_length=400)
    # these[created_on,updated_on] fields are auto creating
    created_on = models.DateField(auto_now_add=True)
    updated_on = models.DateField(auto_now=True)


# class model sessontoken for generationg sesson token
class SessionToken(models.Model):
    user = models.ForeignKey(UserModel)
    session_token = models.CharField(max_length=255)
    last_request_on = models.DateField(auto_now=True)
    created_on = models.DateTimeField(auto_now_add=True)
    is_valid = models.BooleanField(default=True)
```

```
# Post Model for handaling post data
class PostModel(models.Model):
    user = models.ForeignKey(UserModel)
    image = models.FileField(upload_to='user_images')
    image_url = models.CharField(max_length=255)
    caption = models.CharField(max_length=1000)
    created_on = models.DateTimeField(auto_now_add=True)
    updated_on = models.DateTimeField(auto_now=True)
```

5.2.3 View

View is short form of view file. It is a file containing Python function which takes web requests and returns web responses. A response can be HTML content or XML documents or a \404 error" and so on. The logic inside the view function can be arbitrary as long as it returns the desired response. To link the view function with a particular URL we need to use a structure called URL conf which maps URLs to view functions.

```
def login_view(request):
    response_data = {}
    if request.method == 'POST':
        form = SignInForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            user =
UserModel.objects.filter(username=username).first()
            if user:
                # now check for password is true or not
                if check_password(password, user.password):
                    # if password is valid it will create a
sesson token for each user
                    token = SessionToken(user=user)
                    token.create_token()
                    token.save()
                    response = redirect('/feed/')
```

```
                    response.set_cookie(key='session_token',
value=token.session_token)
                    return response
                else:
                    # if password username is valid but password
is not valid
                    wrong_password = "Password or Username is
Invalid. Please try Again !!!"
                    return render(request, 'login.html',
{'color': 'w3-red w3-large', 'status': wrong_password})
            else:
                # if no such username found
                wrong_password = "Username or Password  Is
Invalid. Please try Again !!!"
                return render(request, 'login.html', {'color':
'w3-red w3-large', 'status': wrong_password})

    elif request.method == 'GET':
        form = SignInForm()
    response_data['form'] = form
    return render(request, 'login.html', response_data)
```

5.2.4 Template

Django's template is a simple text file which can generate a text-based format like HTML and XML. The template contains variables and tags. Variables will be replaced by the result when the template is evaluated. Tags control the logic of the template. We also can modify the variables by using filters. For example, a lowercase filter can convert the variable from uppercase into lowercase.

```
<form method="POST" action="/comment/" class="w3-show-inline-
block">
    <li class="w3-show-inline-block">
```

```
          {% csrf_token %}
            <input type="hidden" name="post" value="{{ post.id
}}" >
            <input class="w3-input w3-hover-teal w3-large w3-
padding w3-round" name="comment_text" type="text"
id="comment_text" placeholder="Write a comment..." value="{{
comment_text }}">
          {{ comment_text }}
    </li>
    <li class="w3-show-inline-block">
    <button class="w3-button w3-margin-left w3-margin-right w3-
xlarge w3-circle w3-teal" type="submit" value="Comment"><i
class="fa fa-comment"></i></button>

    </li>
</form>

<ul class="w3-ul w3-card-4">
    {% for comment in post.comments %}
    <li class="w3-bar">

        <img src="/static/Assets/img.png" class="w3-bar-item w3-
circle" style="width:85px">
        <div class="w3-bar-item" style="padding-left: 0px;">
            <p class="">
                <span class="w3-medium w3-bold">{{
comment.user.name }}</span>
                <span class="w3-text-teal w3-margin-left w3-
large">{{ comment.comment_text }}</span>
                <br><span class="w3-small">{{ comment.created_on
}}</span>
            </p>
        </div>
        {% endfor %}
    </li>
</ul>
```

5.3 Introduction to SQLite

SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()

SQLite is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) If optional features are omitted, the size of the SQLite library can be reduced below 300KiB. SQLite can also be made to run in minimal stack space (4KiB) and very little heap (100KiB), making SQLite a popular database engine choice on memory constrained gadgets such as cellphones, PDAs, and MP3 players. There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite be faster than direct filesystem I/O.

SQLite is very carefully tested prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage. SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are

ACID even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides bugs lists and minute-by-minute chronologies of code changes.

The SQLite code base is supported by an international team of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the published interface spec, SQL syntax, and database file format. The source code is absolutely free to anybody who wants it, but professional support is also available.

The SQLite project was started on 2000-05-09. The future is always hard to predict, but the intent of the developers is to support SQLite through the year 2050. Design decisions are made with that objective in mind.

We the developers hope that you find SQLite useful and we entreat you to use it well: to make good and beautiful products that are fast, reliable, and simple to use. Seek forgiveness for yourself as you forgive others. And just as you have received SQLite for free, so also freely give, paying the debt forward.

5.4 Introduction to MySQl

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

5.5 Introduction to HTML

HTML stands for Hyper Text Mark-up Language. Using HTML, we can create a Web Page which can be viewed through a Web Browser over Internet. HTML describes the contents of a web page with Mark-up Tags. Tags are defined by angle brackets <>. <> is called the Opening Tag or the Start Tag. The Closing Tag or the End Tag is defined as </>. The whole HTML document is enclosed in between the opening <html> tag and the closing </html>tag. An HTML file has a .html or .htm extension.

5.5.1 The General Structure of an HTML document

An HTML document is divided mainly into two parts- Head and Body. The Head element contains the Title of the Page which is written in between <title>...</title> tags. The Title of the Page will be shown in the Header of the Web Page.

The second part is the Body Element. This is the most important part of the HTML document. The Body element contains all the contents to be displayed in a web page (texts, images etc.).

5.5.2 Basic Elements of a Body and their Attributes

The following Basic elements are used in a body that describe the contents of the body

- **Headings**
- **Paragraph**
- **Formatting Elements**
- **Image**
- **Anchor**

5.5.3 HTML Headings

In HTML, six types of headings are used, viz, H1, H2, H3, H4, H5 and H6. Headings are important in maintaining the flow of the contents, i.e., the hierarchical structure of the document. H1 is the most important and H6 is the least important heading.

Each heading has an Opening tag and a Closing tag. The use of headings can be shown as below-

<h1>This is the first heading</h1>

<h2>This is the second heading</h2>

<h3>This is the third heading</h3>

<h4>This is the fourth heading</h4>

<h5>This is the fifth heading</h5>

<h6>This is the sixth heading</h6>

5.5.4 Paragraph

Paragraphs in an HTML page are encoded within and tags. However, the end tag is optional. The space between two paragraphs is automatically specified in HTML. We can insert a line break at any point of the document by using a
 tag.

This is my first paragraph.

This is another paragraph. This will give you

an introduction of HTML.

5.5.5 Formatting Elements

There are three basic formatting elements used in HTML to make a text Bold, Italic and Underlined.

To make a text bold, and tags are used.

To make a text italic, and tags are used.

To make text underlined, <u> and </u> tags are used.

5.5.6 Image

Images are often used in a web page. To add an image in HTML, we use the <img> tag. The path of the location of the image is written in the "src" attribute. If the image is located in the same folder as the web page then only the name of the image file is included. We can also format the size of an image by specifying its width & height.

Take the following example:

<img src="roop.gif" width="100" height="80">

5.5.7 The Anchor Element

HTML provides the facility of links in a web page to other web pages. By clicking on some text as a link, we can go to a different web page. This is called Hyper Linking of web pages. This makes a web page dynamic.

The anchor tag <a> is used for this purpose.

text as the link

What is HTML?

HTML is the standard markup language for creating Web pages.

HTML stands for Hyper Text Markup Language

HTML describes the structure of Web pages using markup

HTML elements are the building blocks of HTML pages

HTML elements are represented by tags

HTML tags label pieces of content such as "heading", "paragraph", "table", and so on

Browsers do not display the HTML tags, but use them to render the content of the page

5.6 Introduction to CSS

A CSS (cascading style sheet) file allows you to separate your web sites (X)HTML content from it's style. As always you use your (X)HTML file to arrange the content, but all of the presentation (fonts, colors, background, borders, text formatting, link effects & so on…) are accomplished within a CSS.

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

5.6.1 Internal Stylesheet

First we will explore the internal method. This way you are simply placing the CSS code within the <head></head> tags of each (X)HTML file you want to style with the CSS. The format for this is shown in the example below.

<head>

<title><title>

<style type="text/css">

CSS Content Goes Here

</style>

</head>

<body>

5.6.2 External Stylesheet

Next, we will explore the external method. An external CSS file can be created with any text or HTML editor such as "Notepad" or "Dreamweaver". A CSS file contains no (X)HTML, only CSS. You simply save it with the .css file extension. You can link to the file externally by placing one of the following links in the head section of every (X)HTML file you want to style with the CSS file.

Either of these methods are achieved by placing one or the other in the head section as shown in example below

<link rel=”stylesheet” type=”text/css” href=“Path To stylesheet.css” />

Or you can also use the @import method as shown below

<style type=”text/css”>@import url(Path To stylesheet.css)</style>

```
body{
    background: url("/static/Assets/background (34).jpg") fixed;
    overflow-y: hidden;
    overflow-x: hidden;
}

<!--w3css theme color-->
  <link rel="stylesheet" href="/static/StyleSheet/w3-theme-
teal.css">
  <link rel="stylesheet" href="static/StyleSheet/w3-theme-
teal.css">
  <link rel="stylesheet" href="https://www.w3schools.com/lib/w3-
theme-teal.css">
  <!--cdn-->
  <!--<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">-->
  <!--Animate css connected-->
  <link rel="stylesheet" href="/static/StyleSheet/animate.css">
  <link rel="stylesheet" href="static/StyleSheet/animate.css">
  <!--cdn-->
  <link rel="stylesheet" href="/static/StyleSheet/Style.css">
  <link rel="stylesheet" href="static/StyleSheet/Style.css">

<style>
html,body,h1,h2,h3,h4,h5 {
    font-family: "Open Sans", sans-serif
}

</style>
```

5.7 Introduction to JavaScript

JavaScript is a cross-platform, object-oriented scripting language. It is a small and lightweight language. Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

5.8 JavaScript and Java

JavaScript and Java are similar in some ways but fundamentally different in some others. The JavaScript language resembles Java but does not have Java's static typing and strong type checking. JavaScript follows most Java expression syntax, naming conventions and basic control-flow constructs which was the reason why it was renamed from LiveScript to JavaScript.

JavaScript is a very free-form language compared to Java. You do not have to declare all variables, classes, and methods. One of many JavaScript HTML methods is getElementById().This example uses the method to "find" an HTML element (with id="demo") and changes the element content (innerHTML) to "Hello JavaScript":

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

5.9 Introduction to Bootstrap

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for common user interface components like Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel and many other as well as optional JavaScript extensions.

Bootstrap also gives your ability to create responsive layout with much less effort

5.9.1 Advantages of Bootstrap

The biggest advantage of using Bootstrap is that it comes with free set of tools for creating flexible and responsive web layouts as well as common interface components.

Additionally, using the Bootstrap data APIs you can create advanced interface components like Scroll spy and Type heads without writing a single line of JavaScript.

5.9.2 Bootstrap Package Includes

Components - Bootstrap provides built-in reusable components that provide alerts, navigation, iconography, dropdowns etc.

JavaScript Plugins - Bootstrap provides many custom jQuery plug-in. You can use them to increase functionality.

CSS - Bootstrap provides extensible classes, CSS setting including advanced grid system. Customization - Bootstrap allows you to customize Bootstrap components and jQuery plug-in to build your own version.

5.9.3 Bootstrap CDN

<!-- Latest compiled and minified CSS -->

<linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

5.10 Introduction to Font Awesome

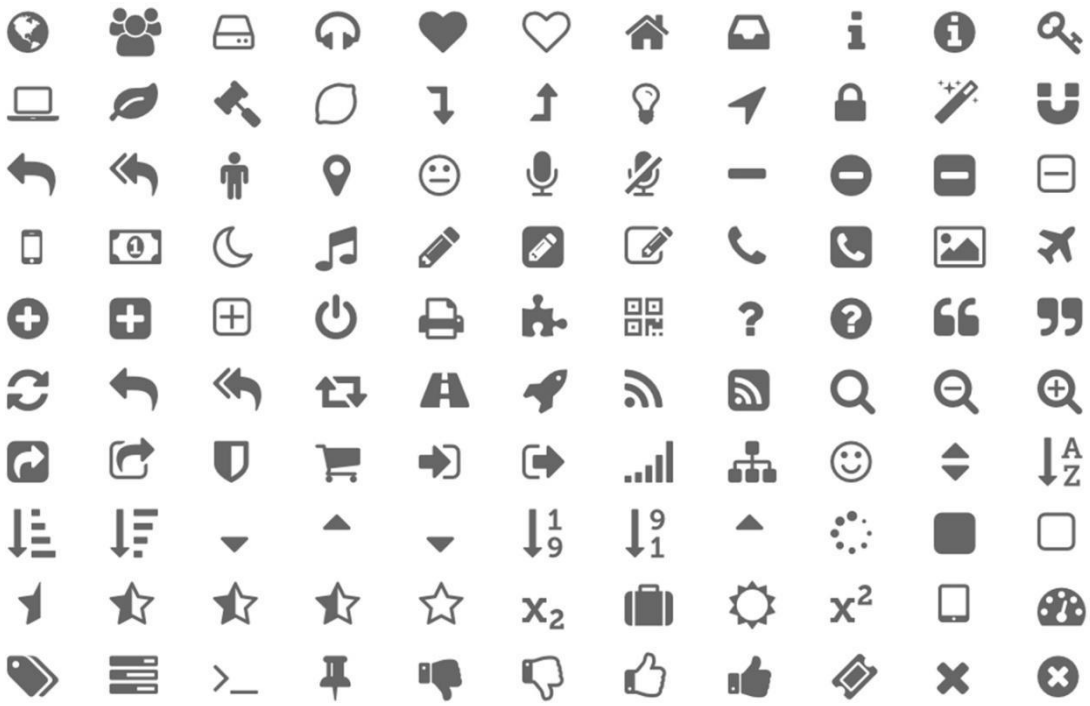
Font Awesome is a font and icon toolkit based on CSS and LESS. It was made by Dave Gandy for use with Twitter Bootstrap, and later was incorporated into the Bootstrap CDN. Font Awesome has a 20% market share among those websites which use third-party Font Scripts on their platform, ranking it second place after Google Fonts.

Font Awesome gives you scalable vector icons that can instantly be customized — size, colour, drop shadow, and anything that can be done with the power of CSS.

5.10.1 Font Awesome v4.7.0

Font Awesome is a full suite of 675 pictographic icons for easy scalable vector graphics on websites, created and maintained by Dave Gandy. Stay up to date with the latest release and announcements on Twitter: @fontawesome.

Get started at http://fontawesome.io!



One Font, 675 Icons

In a single collection, Font Awesome is a pictographic language of web-related actions.

No JavaScript Required

Fewer compatibility concerns because Font Awesome doesn't require JavaScript.

Infinite Scalability

Scalable vector graphics means every icon looks awesome at any size.

Free, as in Speech

Font Awesome is completely free for commercial use. Check out the license.

CSS Control

Easily style icon colour, size, shadow, and anything that's possible with CSS.

Perfect on Retina Displays

Font Awesome icons are vectors, which mean they're gorgeous on high-resolution displays.

Plays Well with Others

Originally designed for Bootstrap, Font Awesome works great with all frameworks.

Desktop Friendly

To use on the desktop or for a complete set of vectors, check out the cheat sheet.

Accessibility-minded

Font Awesome loves screen readers and helps make your icons accessible on the web.

5.10.2 Linking of Font-Awsome

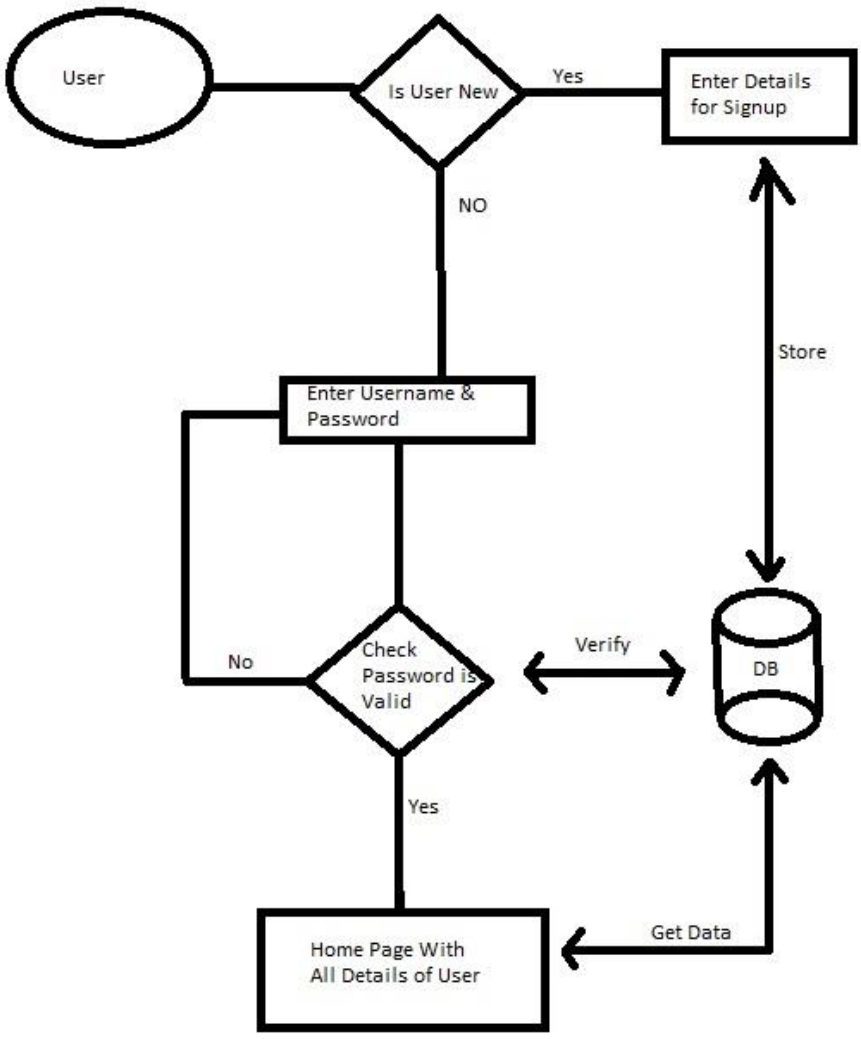
```
<!--font-awsome linked here-->
  <link rel="stylesheet" href="/static/font-awesome-
4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="static/font-awesome-4.7.0/css/font-
awesome.min.css">


<p><i class="fa fa-envelope fa-fw w3-margin-right w3-text-
theme"></i> {{ Current_user.email }}</p>
<p><i class="fa fa-user-secret fa-fw w3-margin-right w3-text-
theme"></i> {{ Current_user.username }}</p>
<p><i class="fa fa-clock-o fa-fw w3-margin-right w3-text-
theme"></i>{{ time.now }}</p>
```

Chapter 6 SYSTEM DESIGN

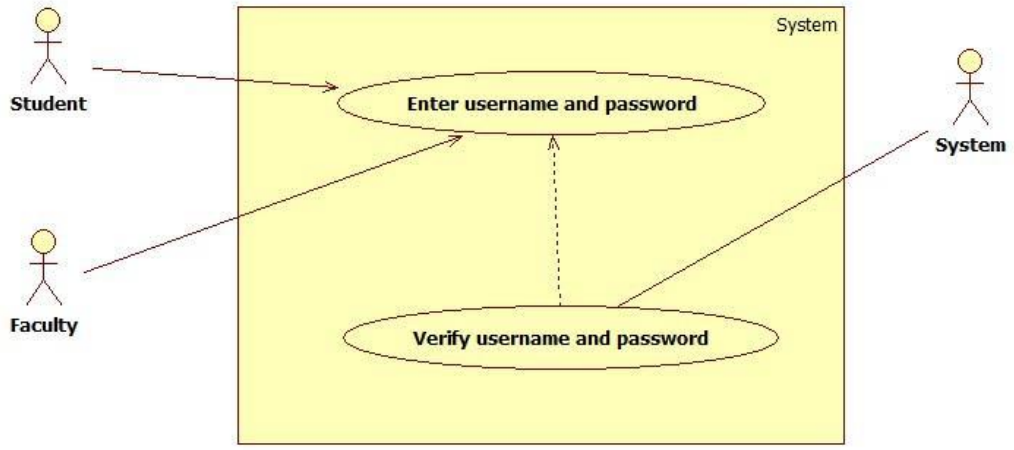
6.1 Introduction:

Design is the abstraction of a solution; it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to create the implementation. In this chapter we are introduce context diagram, models, system architecture, principal system object, design model and object interface.

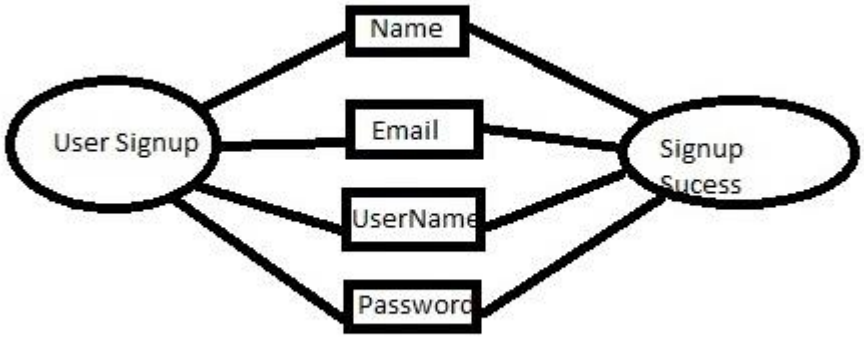
6.2 ER- Diagram:



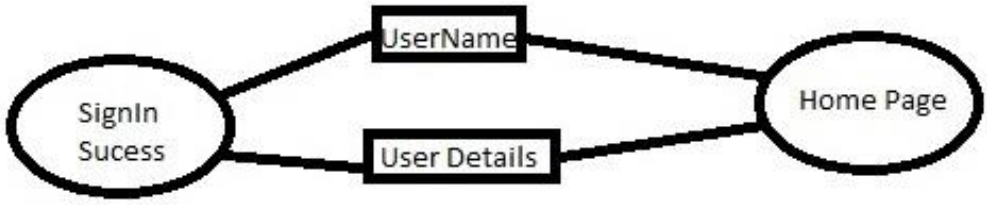
6.3 Use Case Diagram:



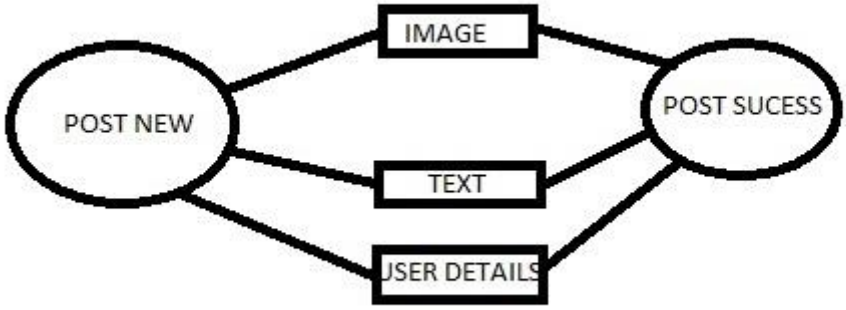
6.3.1 User Sign



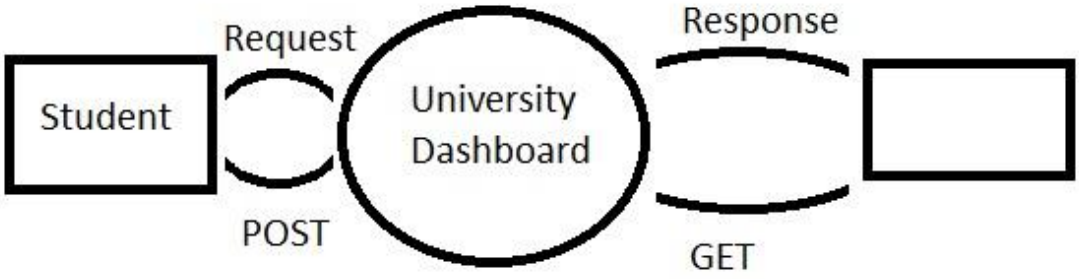
6.3.2 User Login



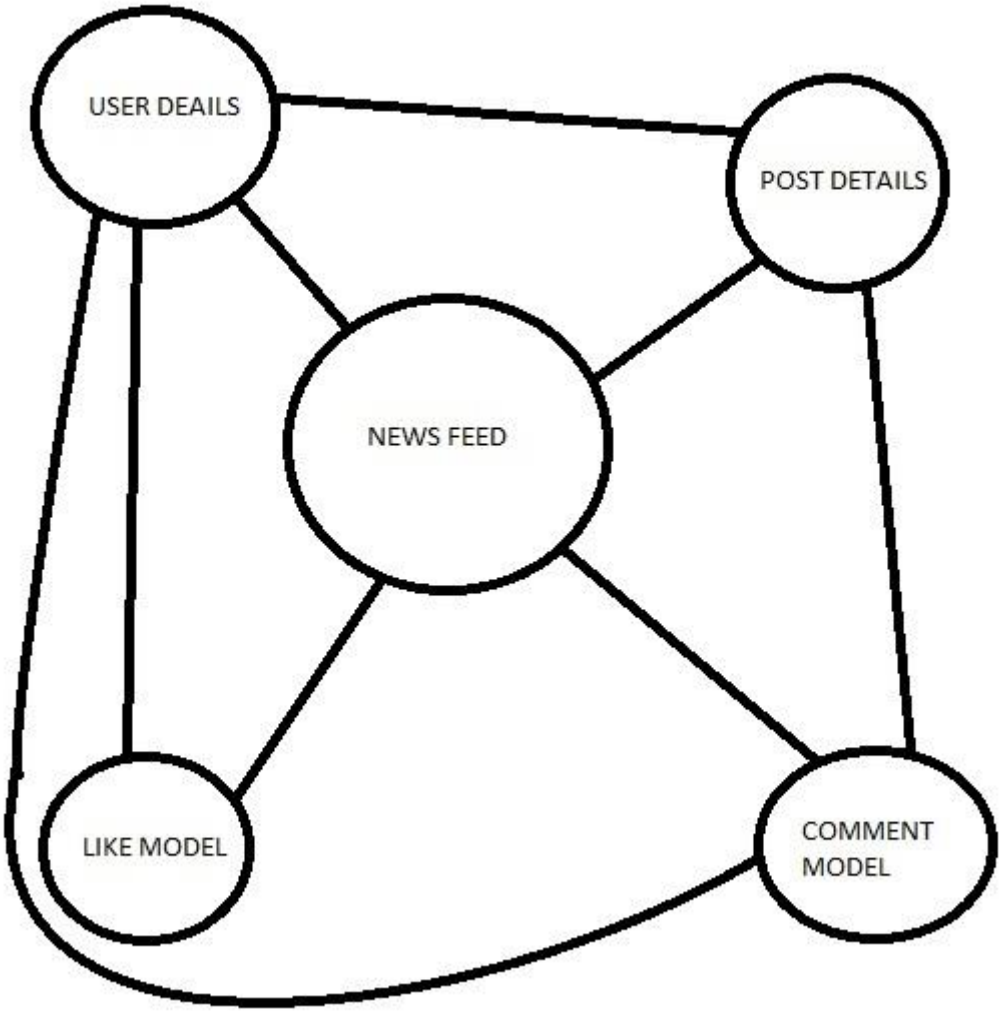
6.3.3 POST Status



6.4 Zero Level Data Flow Diagram(DFD)



6.5 One-Level Data Flow Diagram(DFD)



Chapter 7 FURTHER ENHANCEMENTS

The Scope of this project is that it could be used in Our University ERP integrated with it and make our ERP system more interesting for everyone to use it and share useful information with all the member of origination.

In this project its totally depend on the university or any specific origination and to use this software it provides the main and must basic task to the end user as service to help them for easily access to the information regarding their organization.

It could be used in any origination as well and work with that origination internal member and also user are able to share their idea and according to which the other member of origination can show interest and comments there view on that idea.

This project would be very useful for educational institutes where regular evaluation of students' is required. Further it can also be useful for anyone who requires feedback based on objective type responses.

Chapter 8 PROJECT OVERVIEW

8.1 An overview of the Django framework development process

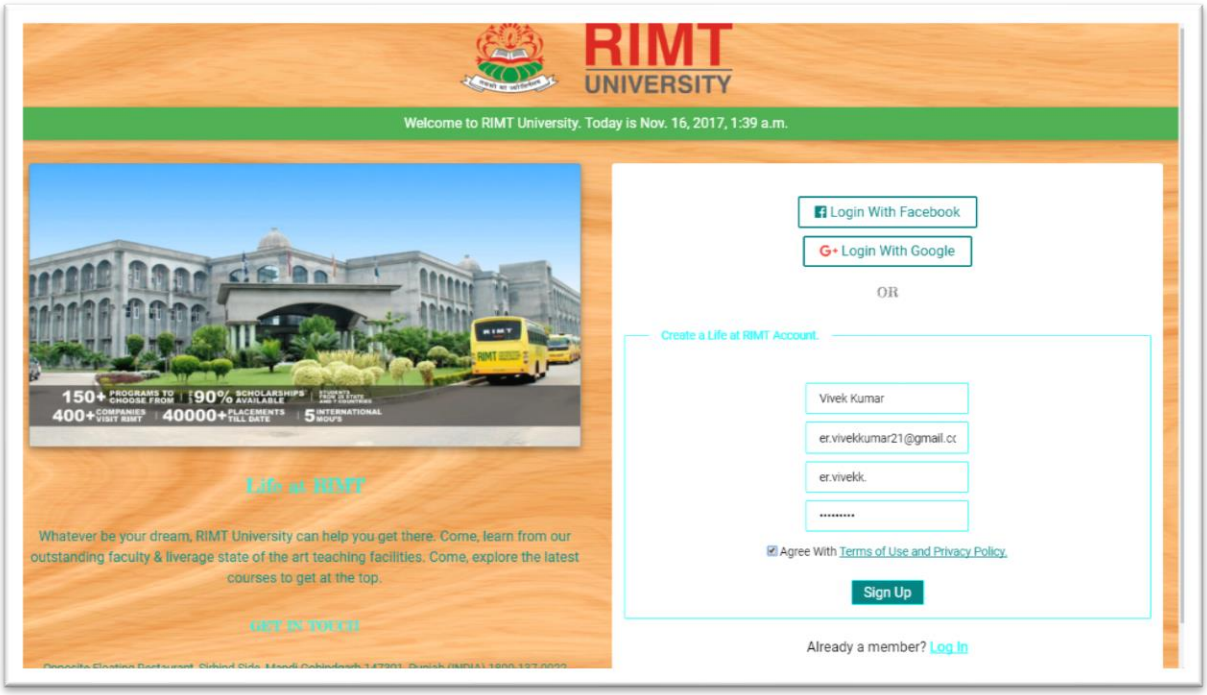
To build such a complicated web system, we need three major parts for each component: database, user interface and the functions to interact in between. Django framework provides sufficient functionalities to implement these three parts.

Corresponding to database, user interface and functions in between, Django has model, template and view components to deal with each part respectively. Django's model component helps programmer to define and maintain tables in the database, while its template component helps to write html files using a combination of both html syntax and Django syntax. For those functions in between, Django provides a view component which reads the input from user interface and makes corresponding changes in the database.

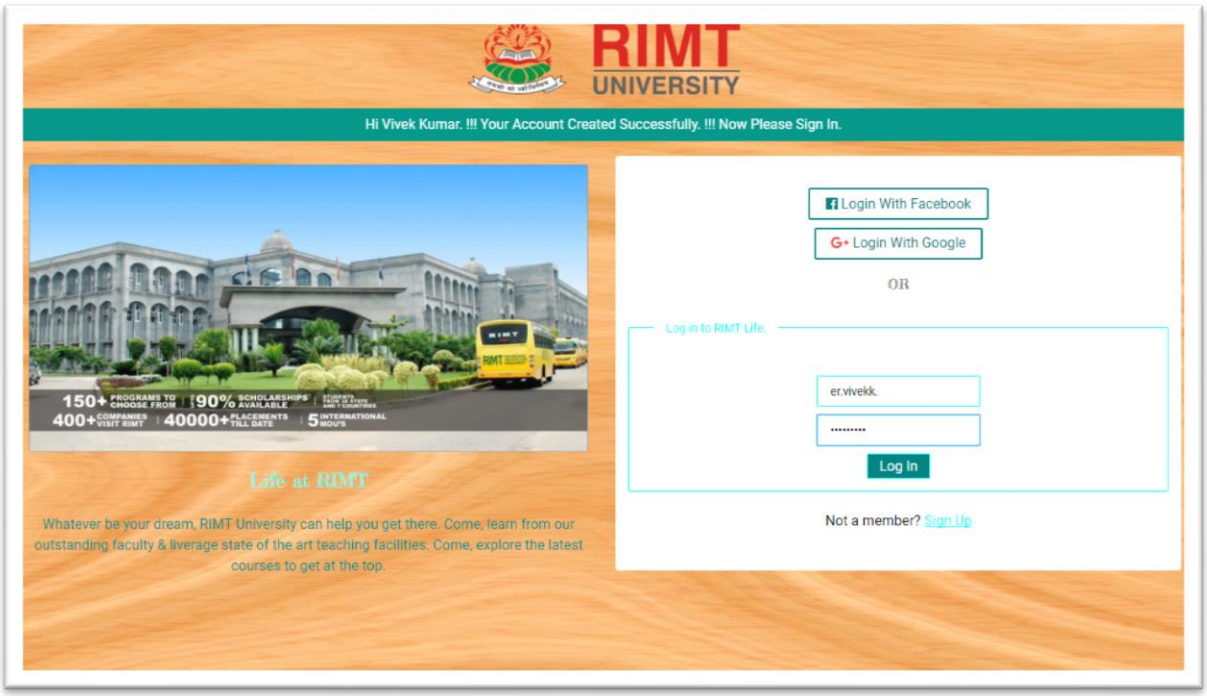
So far, we have our backend database and the frontend web page user interface. What we need now is the logic in between to deal with the user requests and maintain the database. Django view component provides a set of application programming interfaces to fulfil our need and help us implement the logic.

The Django view file is where we write our function to achieve the above two goals. First, it is used to pass parameters to the template and call the right template for the user. Every time we input a URL in the address bar or click a hyperlink in the system, Django will call the right view function based on that URL. Then the function will return a template as well as the corresponding parameters. Thus we can see the actual web page displaying the information we need. Second, if we submit something such as create group, the function will have an http request as its input parameter. Based on that parameter the database is updated or the user is provided the required information. The view function for creating a group is given below:

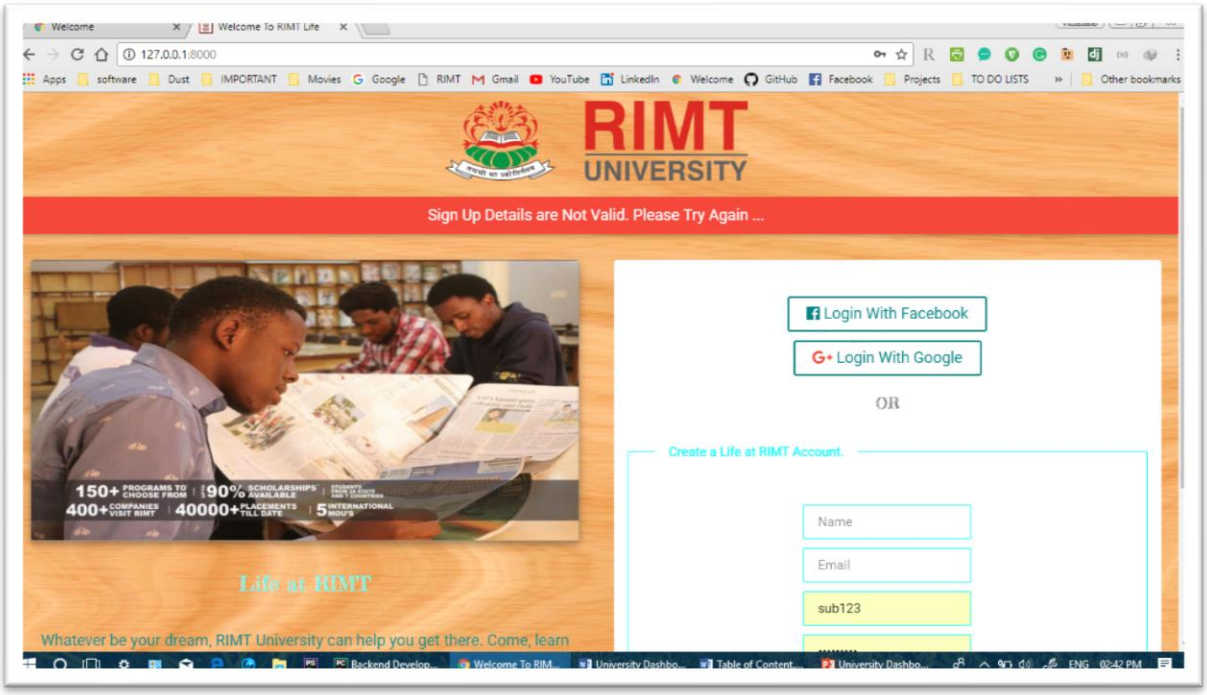
8.2 UI of the Initial Signup Application



8.3 UI of Sign Up Success



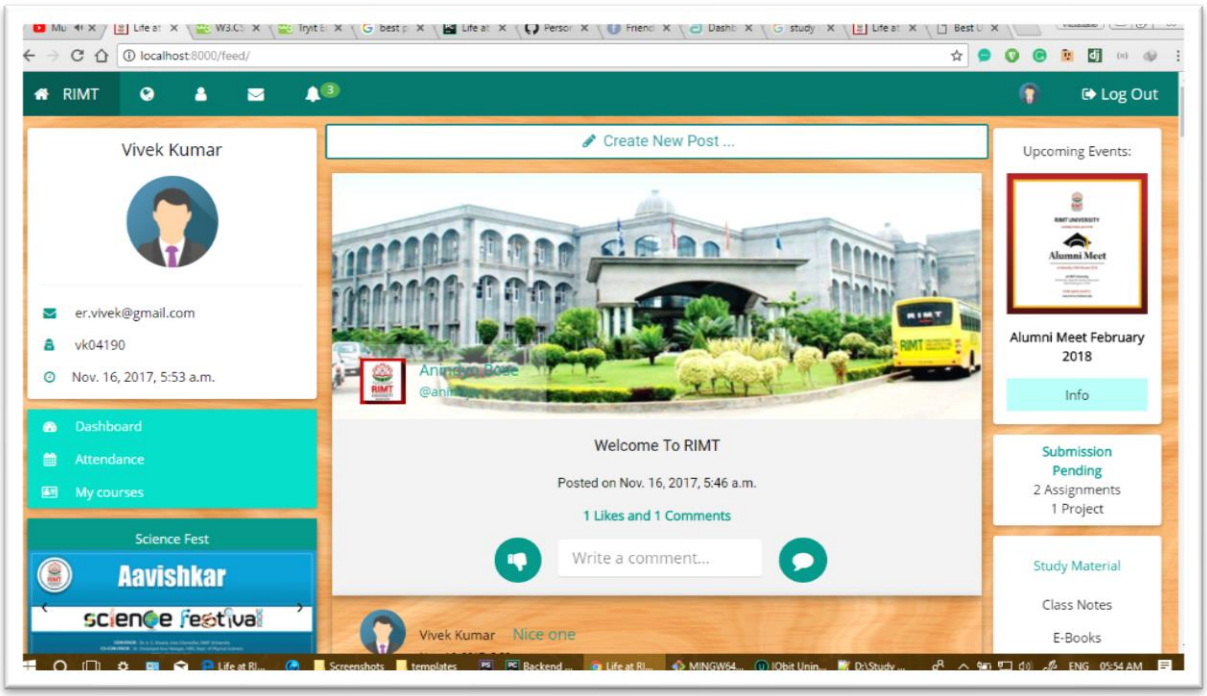
8.4 UI of Signup Failure On Wrong Data



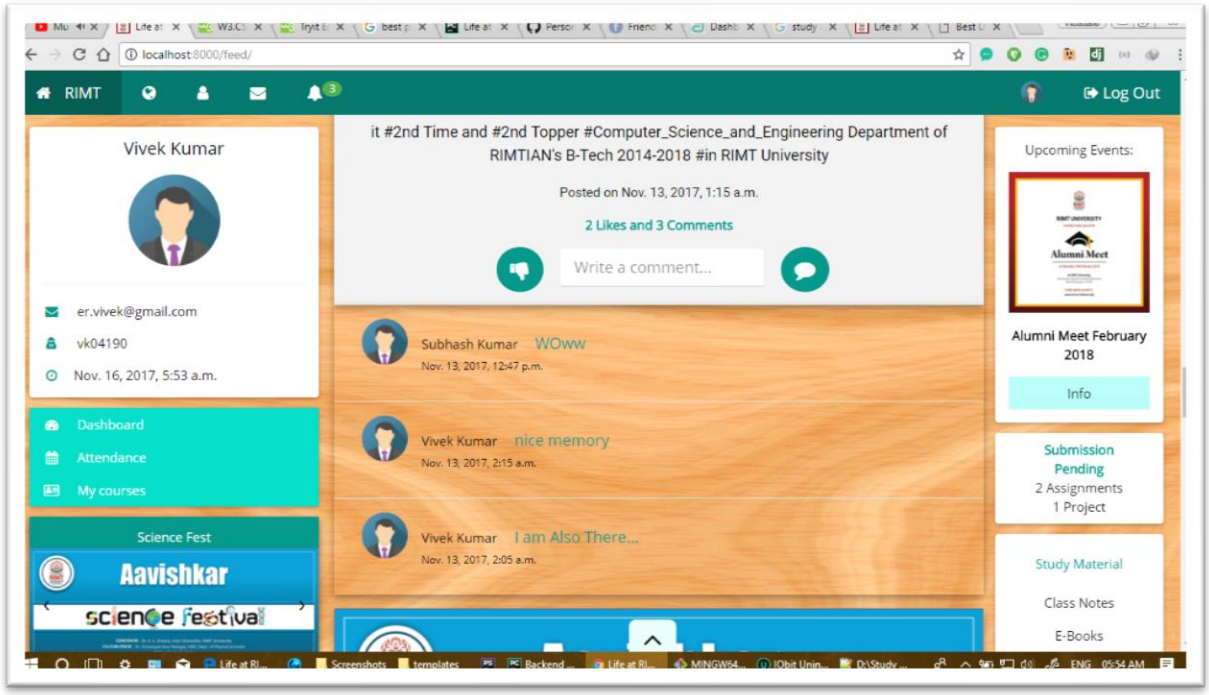
8.5 UI of Login Page

8.6 UI of Wrong Password

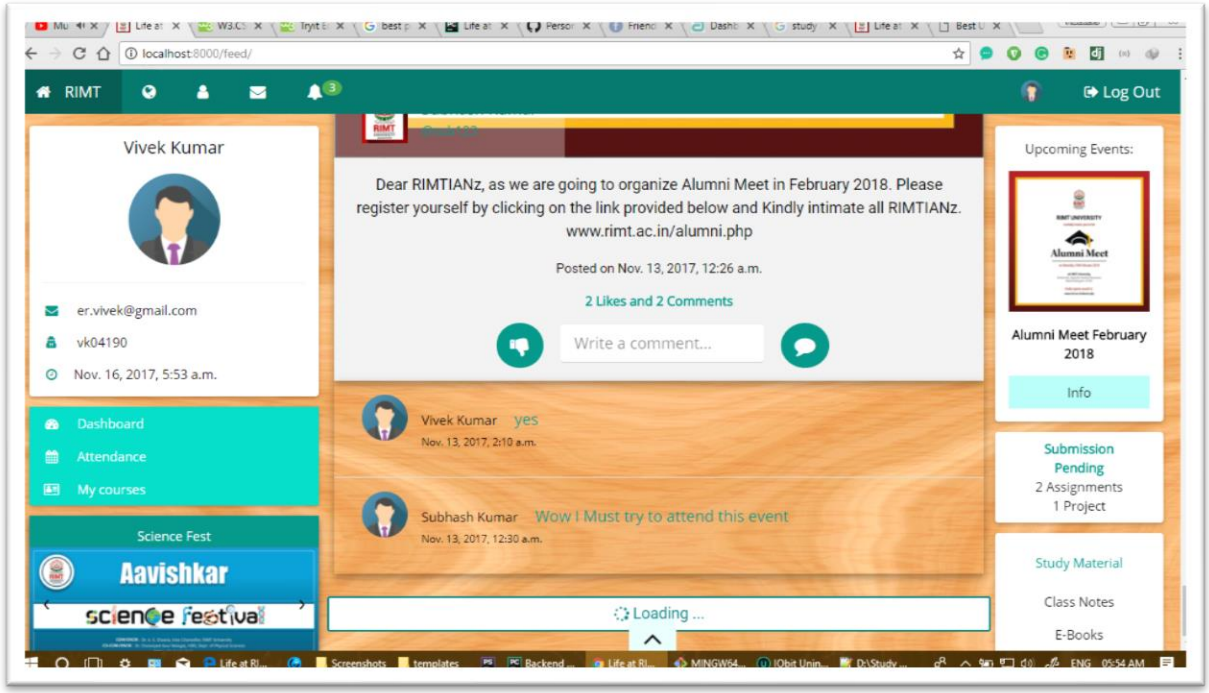
8.7 UI of Initial Feed Page



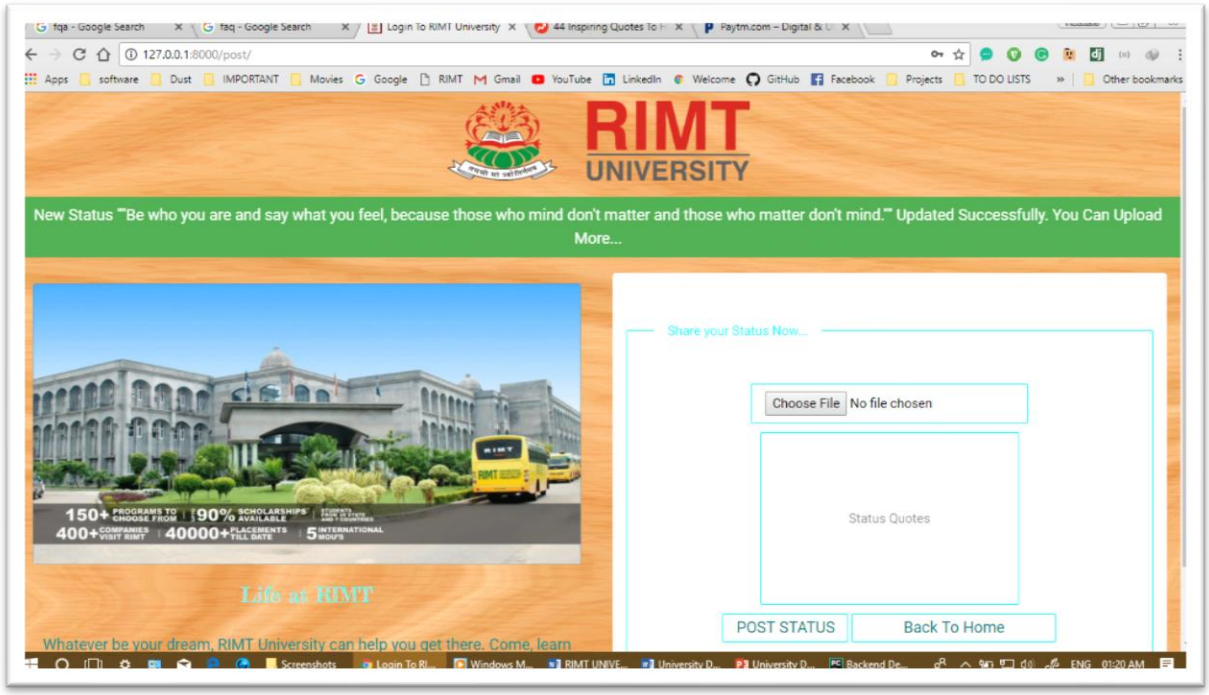
8.8 UI of Like and Comment Section



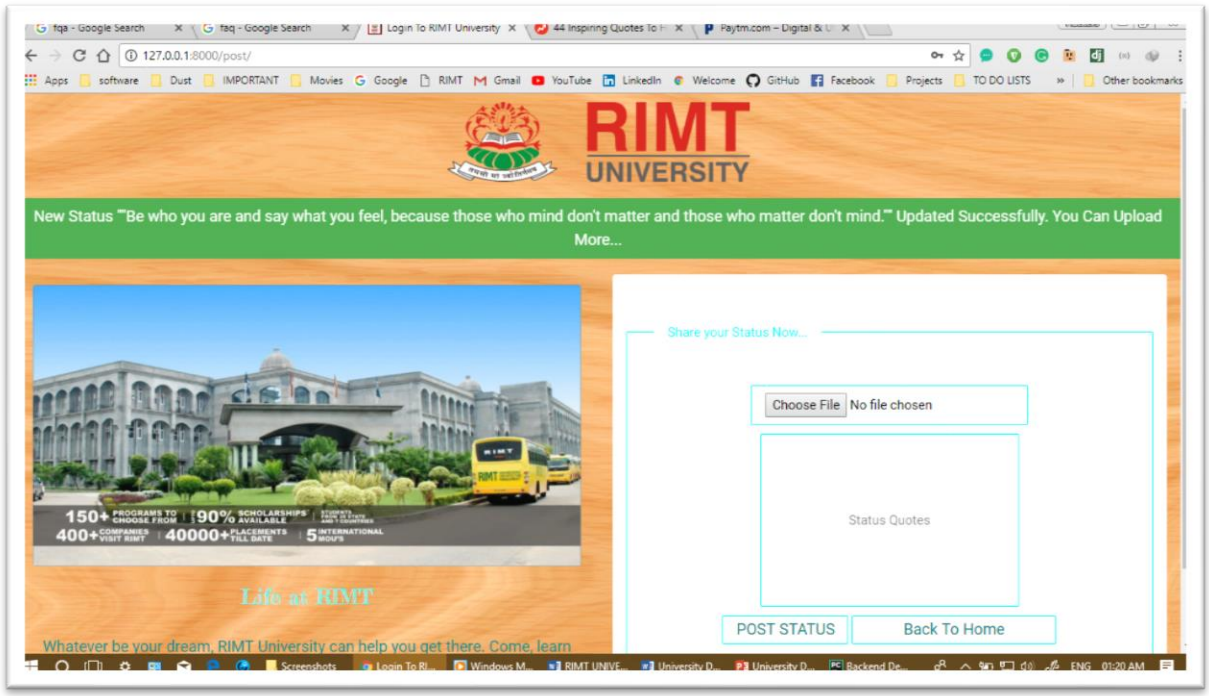
8.9 UI Of News Feed Loading More …



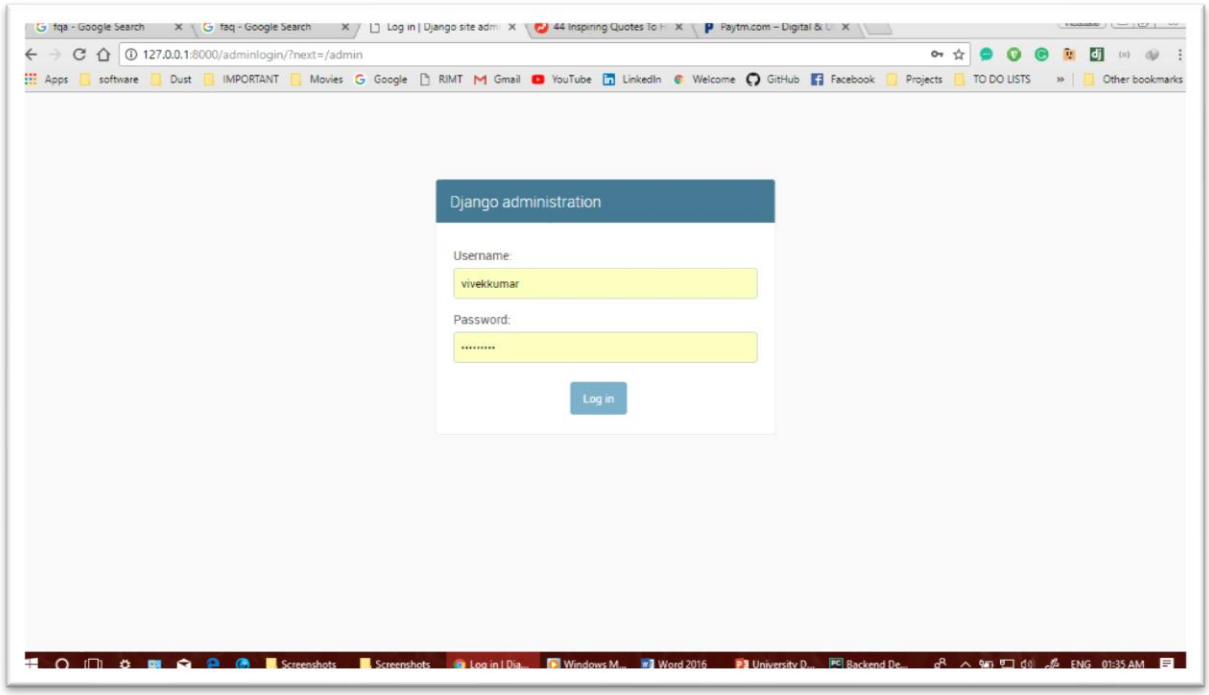
8.10 UI of Upload a New Post



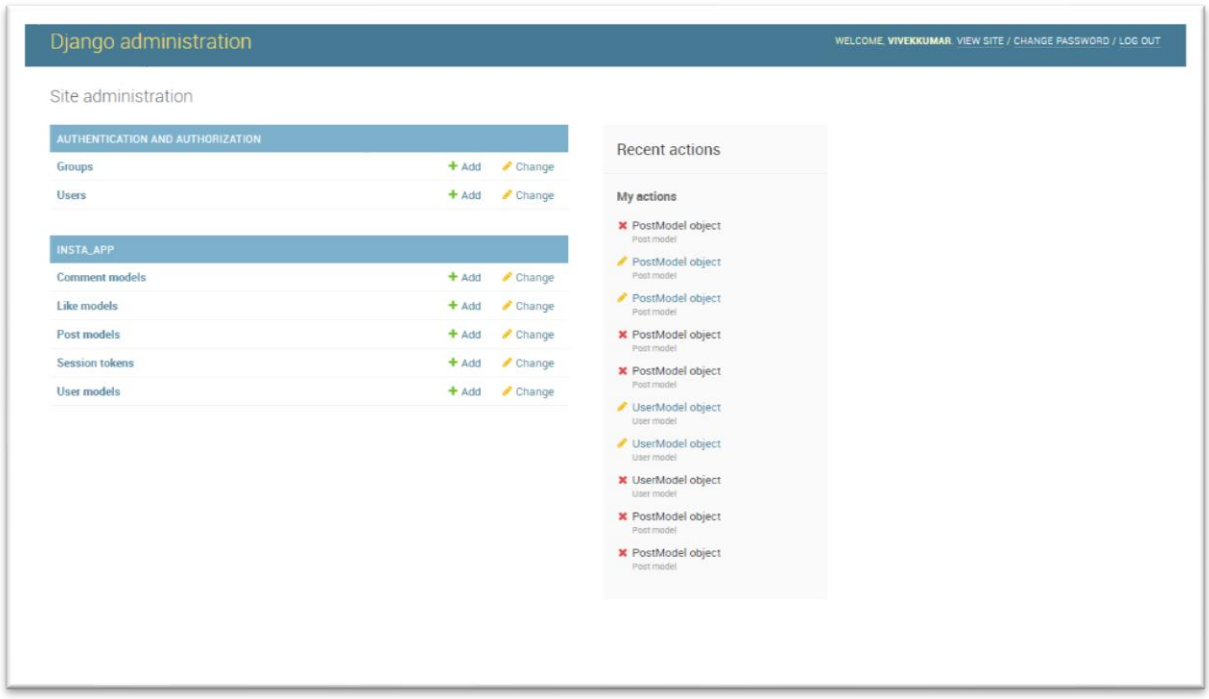
8.11 UI OF POST SUCCESS



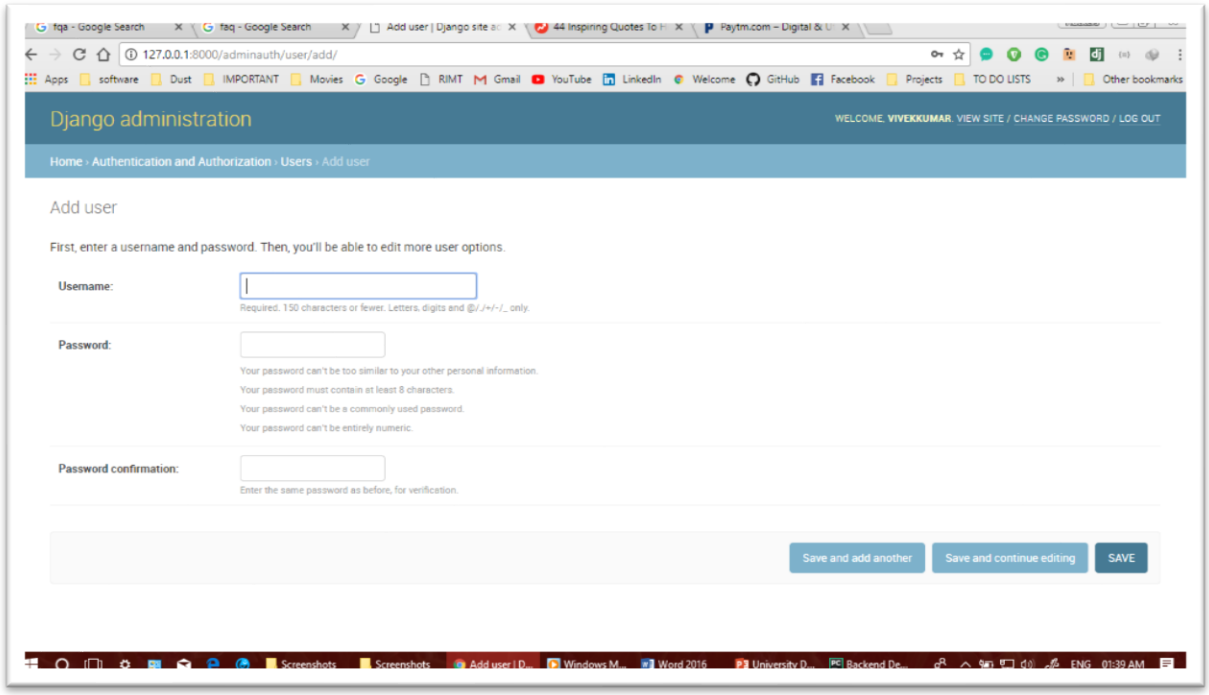
8.12 UI of Admin Login Page



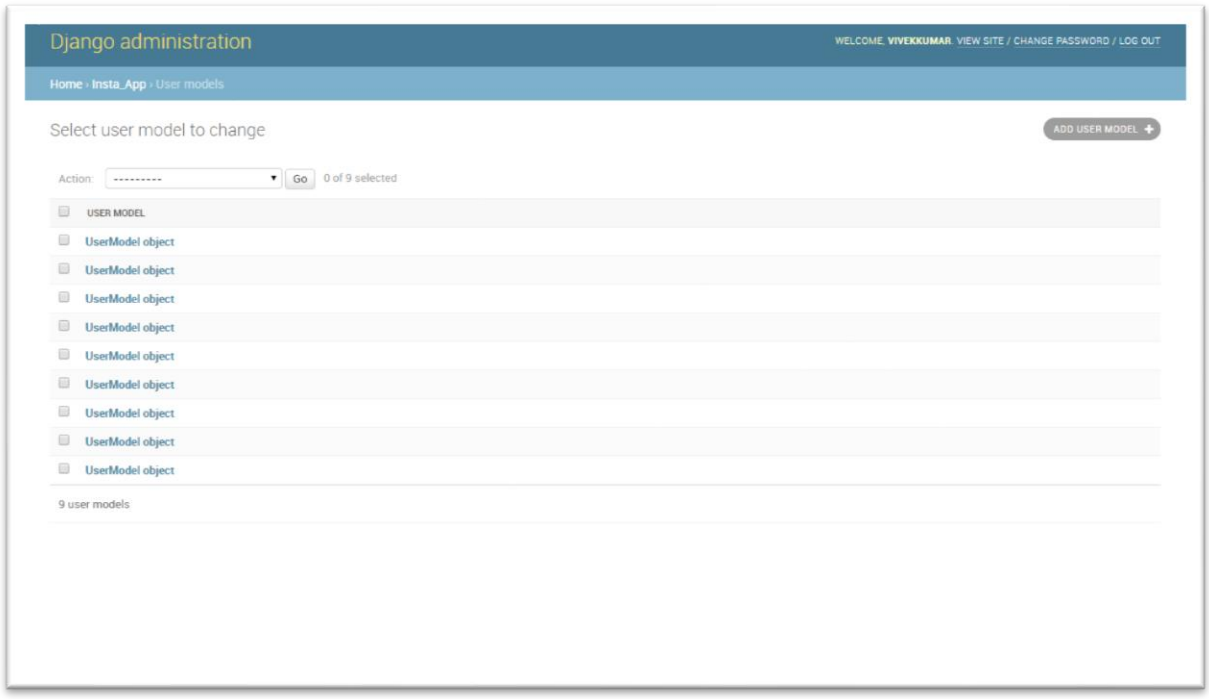
8.11 UI of Admin Control Panel



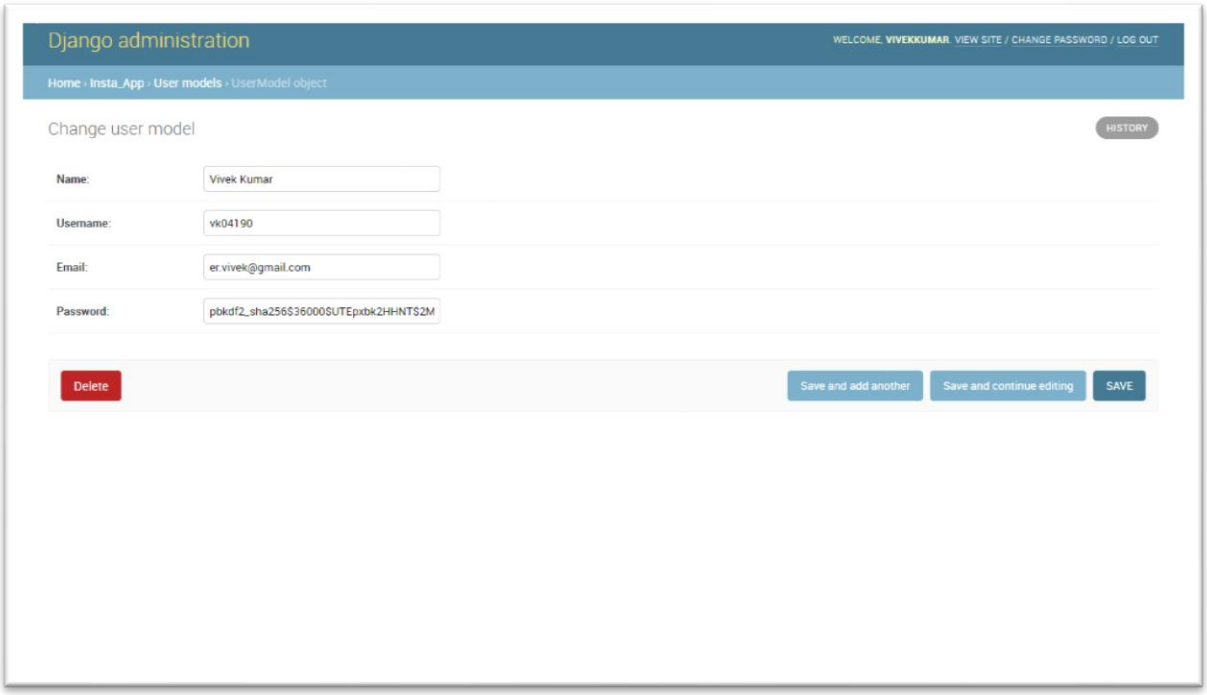
8.12 Admin UI of Create New Admin



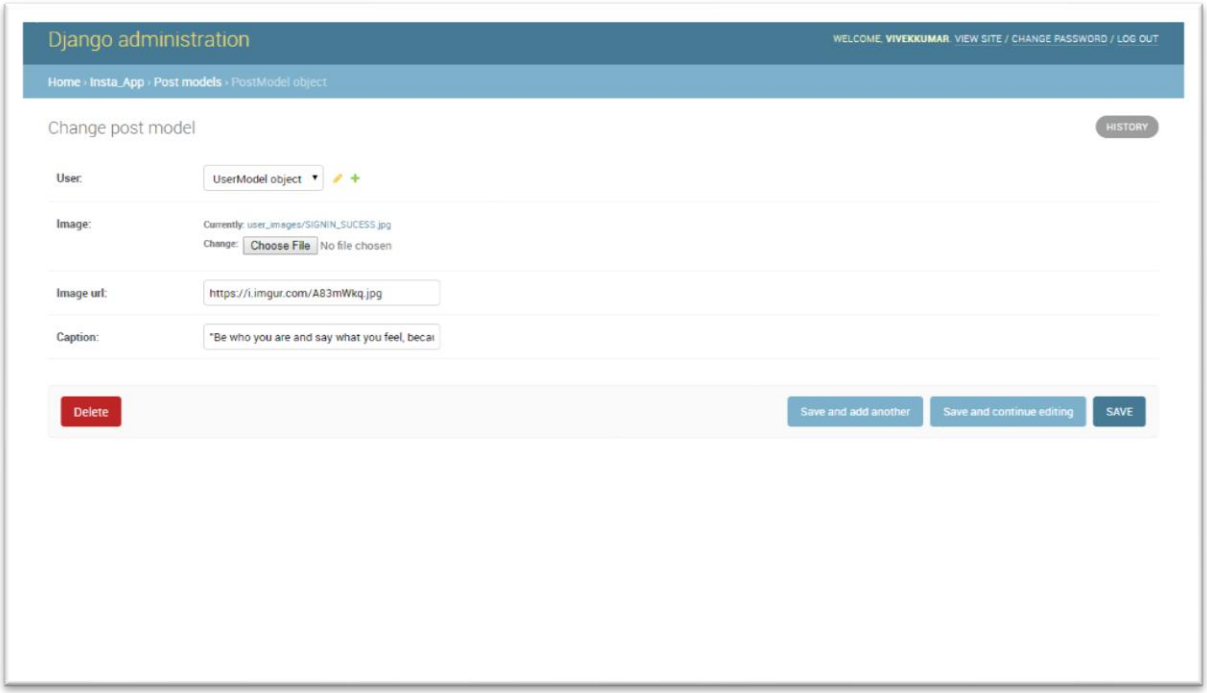
8.13 Admin UI of Admin All User Model



8.14 Admin UI of Specific User



8.15 Admin UI of Specific Post Model



Chapter 9 CONCLUSION

Project “RIMT University Dashboard” Is Basically Made for a new Module of Current ERP of our University which make possible to provide a system with a good user interface for user to interact with the academic’s system to view their data regarding University, like Attendance, Projects, Assignment, Hostel Faculty, Bus facility, and these all will have integrated with a new module called University Dashboard where each and every user of University

Dashboard user able to create a specific account for them and after that they will redirected to login page where they will login to their account and after that the user redirected to the Core ERP home page and there he/she can view there academics details like attendance, Course he/she register for , Pending project or assignment if any, upcoming event in university or any holidays. And in the canter of the home page you will find a Real Time News feed where they can post their own pictures and Regarding their status and after there every member of the university can view that posts and Like, Comments on that post and Many More.

The Django framework gives us a simple and reliable way to create the course University Dash. It provides powerful functionalities and concise syntax to help programmers deal with the database, the web page and the inner logic. The experience of developing the group component in the system also helped me learning a lot of website development with Django. Within the Django framework, we have successfully accomplished the requirements of the system. Once this system passes the testing phase, it can be used to serve students and instructors and substitute several systems currently in service. It will make the work for instructors to manage the course much easier. It also can simplify the operations for students with grade book, submission, and group management all in one system. In short, this system will bring great user experience to both instructors and students. The only limitation for this course system is that although the developers have been testing it with various use cases, it may still encounter problems during real time use. However, even if that happens, the flexibility of Django would provide a simple way to fix the problem, as well as add new features into the system.

Chapter 10 REFRENCES

[1] Django homepage. http://www.djangoproject.com/.

[2] Python documentation. http://www.python.org/doc.

[3] Django (web framework). http://en.wikipedia.org/wiki/Django.

[4] Python (programming language). http://en.wikipedia.org/wiki/Python.

[5] Learn HTML Code, Tags & CSS: https://html.com/

[6] JavaScript (programming language): https://www.w3schools.com/js/

[7] Font Awesome, the iconic font and CSS toolkit: http://fontawesome.io/

[8] Bootstrap- The most popular CSS, and JS library in the world: https://getbootstrap.com/

[9] W3.CSS Templates: https://www.w3schools.com/w3css/w3css_templates.asp

[10] Learn Python the Hard Way: https://learncodethehardway.org/python/