

**MINI PROJECT
(2020-21)**

DOCKERISING A DJANGO WEB APP



Institute of Engineering & Technology

Team Members

SHRUTI SHARMA

171500332

SAHIL SRIVASTAVA

171500281

Supervised By

Mr. Mandip Singh

Asst. Professor

Department of Computer Engineering & Applications

ACKNOWLEDGEMENT

It is our pleasure to acknowledge the assistance of a number of people without whose help this project would not have been possible.

I am highly indebted to **Prof. Dr. Anand Singh Jalal & Mr. Mandip Singh** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents and members of **GLA University** for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

SAHIL SRIVASTAVA

SHRUTI SHARMA

Abstract

The Django Web Application named “Blog” is a blogging website where different users can write different posts. The web app contains an authentication system where a user or an admin can log in using his/her credentials and log out. The web app also contains a registration system where a new new user can register on the website. The login has a forgot password link where a user can reset his/her password by using his email. The user also has the ability to delete and update their post. They can see other posts across the web app, save them, like them and comment on them too.

The web app has admins of the application too. They are ones who handle the web app, they are allowed to view different posts from user accounts, suggest them changes of some sort. If some offensive posts are uploaded, they are allowed to ask them to delete and delete them after a certain period of warning, and are also allowed to register and remove user account.

This web app is deployed on docker container. Docker is an open source project that automates the deployment of applications inside software containers. Docker containers wrap up a piece of software in complete filesystem that contains everything it needs to run: code, runtime, system tools, and system libraries-anything you can install on a server. Docker provides an additional layer of abstraction and automation of operating system level virtualization on Windows and Linux. Docker uses the resource isolation features of the Linux kernel such as groups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent ‘containers’ to run within a single Linux Instance, avoiding overhead of starting and maintaining virtual machines. The objective of proposed study is to deploy the containers on Docker. In that one container is a webserver which has a web application which may be of any language such as java, python etc. and another container consists of database of application. We had to connect these two containers and access the application of the first container in third container with databases connectively

Contents

| | |
|-------------------------------------|----|
| ACKNOWLEDGEMENT | 2 |
| Chapter 1 | 6 |
| Introduction..... | 6 |
| 1.1 Overview and Motivation | 6 |
| 1.3 Objective | 7 |
| 1.4 Overview:..... | 7 |
| Chapter 2 | 8 |
| Software Requirement Analysis | 8 |
| Chapter 3 | 14 |
| Software Design..... | 14 |
| Logical Database Requirements | 15 |
| Sequence Diagrams..... | 19 |
| Chapter 4 | 24 |
| Users Tab: | 30 |
| Chapter 5 | 32 |
| Software Testing | 32 |
| 5.1 Software Testing | 32 |
| 5.2 Types of Testing | 32 |
| 5.2.1 Unit Testing | 32 |
| 5.2.2 Integration Testimg..... | 32 |
| 5.2.3 Regression Testing..... | 32 |
| 5.2.4 System Testing..... | 33 |

| | |
|-------------------------------|----|
| 5.2.5 White Box Testing | 33 |
| 5.2.6 Black Box Testing..... | 33 |
| 5.2.7 Alpha Testing | 33 |
| 5.2.8 Beta Testing | 33 |
| 5.3 Test Cases | 34 |
| Chapter 6..... | 37 |
| Chapter 7..... | 38 |

Chapter 1

Introduction

1.1 Overview and Motivation

Minimal Make files:

When you create an application you want to be able to create make files to standardize and automate the way you can compile and build the software. As the make files grow they become very complicated and unmanageable. With multi-stage docker you can split and embed most of the functionality in docker file keeping your Make files clean and readable.

The motivation behind the project is that in today's world blogging systems have been a huge boon to the people across the globe where different people can share in their views regarding a topic without any hesitation .the ideas can be negative as well as positive so In this project we have tried to add the feature of blocking a blogger who has posted a negative blog. Further we have deployed the website over docker container so that the blogging website could be viewed easily as it gets converted into an docker image.

My dev environment is portable:

Now if I create my dev environment as a docker image, I can use it anywhere I want by just doing “docker run”. I do not care if I am running on mac/windows/Linux.

1.2 Features

The BLOG app is blogging website for sharing different posts, like and comment them, etc. It is a very user friendly web app. What another main feature it supports is that it has been deployed on docker container instead of virtual machines. This deployment serves a purpose as dockerising a web app can help it work and execute faster as the Docker daemon communicates directly with the host operating system and knows how to ration out resources for the running Docker containers. It's also an expert at ensuring each container is isolated from both the host OS and other containers.

In the real world, it means instead of having to wait a minute for a virtual machine to boot up, you can start a docker container in a few milliseconds

1.3 Objective

The aim for designing this docker blogging web app was to make it work fast and efficient. Docker is used to build, ship, and run distributed applications. You can configure your service instances to run in Docker containers.

A Docker container allows an application to be packed together with its dependencies into a portable virtual package that can run with multi-platform support, isolation, and resource limits applied.

The main benefits of running a service instance in a Docker container are as follows:

Deployment-Application can be packed together to a portable Docker image that can run on different platforms.

Isolation-Containers isolate applications from each other and the underlying infrastructure while providing an added layer of protection for the application.

1.4 Overview:

This is a working document and, as such, is subject to change. In its initial form, it is incomplete by definition, and will require continuing refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of the University Academic Portal.

Chapter 2

Software Requirement Analysis

Modules Description

These are some Modules description of Django BLOG web app

i Create New Account:

A customer who having the account in the world can create a virtual account through this module. This module receives the customer profile details along with their password.

ii Login:

Virtual account holders can login in to the system using this module. Thus this is the secured login page for the customers in the website.

iii Reset password:

This module helps the user to reset their password for their account.

iv Updating/Deleting:

This module is works for admin side mostly. The admin can update some account posts and delete account of some users for some offensive posts and comments after a certain period of warning

There are other features and actions that can be performed on the web app but we are not going to look at all of them at their entirety only the basics. The purpose of this whole exercise is to show the usefulness of object oriented programming .

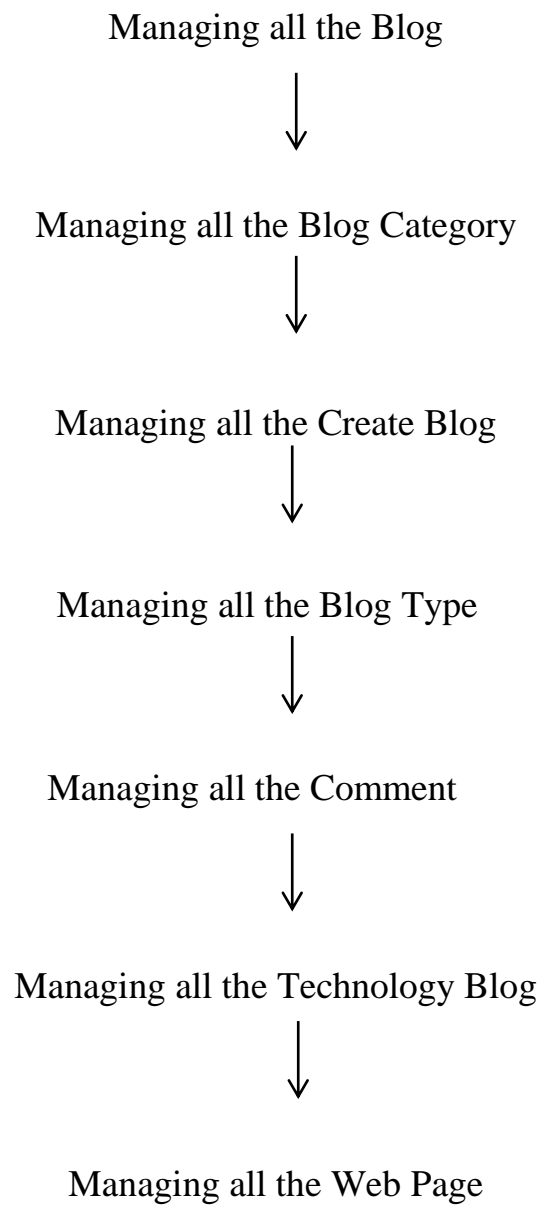


Fig 2.1: Flow Chart Of Blog Web App

Methods

- We need to register and login to account
- We need to post on the app
- We need to update/delete the account

The next thing we need to look at is where to store the information about the account. Obviously, best place is a database. To work with a database will require the following methods:

- Connecting to the database
- Inserting account details

Administrative Modules

Here in my project there are two types of modules. This module is the main module which performs all the main operations in the system.

Admin Module

Admin can access their account as well as the user account and make changes as per the requirement.

- Admin login
- Add/delete/update their account and user accounts too
- Account Details
- User account list
- Active/Inactive account
- View posts, comments and details

User Module

A simple user can access their account and can post of their interests from their account. User can also like, comment and share posts.

- User login, use PIN system
- Creating/open new account registration
- User account details
- Change Password and PIN
- View about account details

Hardware Requirements Specification

The web application will be hosted on a web server which is listening on the web standard port, port 80.

Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus.

Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

Software Requirements Specification

Server side

An Apache web server will accept all requests from the client and forward it accordingly. A database will be hosted centrally using MySQL.

Client side

An OS which is capable of running a modern web browser which supports JavaScript and HTML5 and Django

Technology Used

Django offers multiple options for developing back-end. It is a standard Python interface shipped with Python.

Chapter 3

Software Design

System Design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel.

Database details:

- ✓ The details of Blog is store into the Blog tables respective with all tables.
- ✓ Each entity(Technology Blog, Create Blog, Comment, Blog Category, Blog) contains primary key and unique keys
- ✓ The entity Create Blog, Comment has binded with Blog, Blog Category entities with foreign key
- ✓ There is one-to-one and one one-to-many relationships available between Comment, Blog Type, Technology Blog, Blog
- ✓ All the entities Blog, Comment, Create Blog, Technology, Blog are normalized and reduce duplicacy of records

Design Constraints

The communication between the portal software and the database will be in SQLite portal layout will be produced with HTML/CSS.The product will be written in PHP.The output must be compatible with W3C XHTML 1.0

The source code must follow the coding conventions of PHP.System administrators must have access to comprehensive documentation.

Logical Database Requirements

All data will be saved in the database: user accounts and profiles, discussion data, messages etc. (except files which are stored on the disk.) The database allows concurrent access and will be kept consistent at all times, requiring a good database design.

Data flow diagram:**1) Zero-Level DFD:**

- ✓ Basic overview of the whole online Blog system where processes being analyzed or modeled
- ✓ Designed to be an at-a-glance view of Comment, Technology, Blog and Web page showing the system as a single high-level-process
- ✓ Elaborated the high level proces of Online Blogging

Shows the relationships of Comment, Technology, Blog and Web Page to external entities of Blog, Blog Category and Create Blog

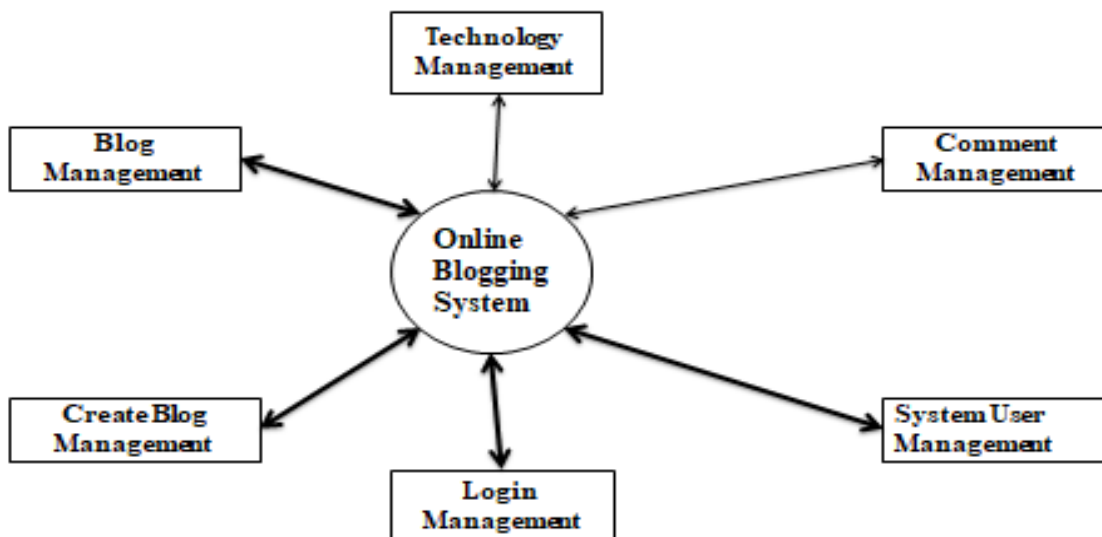


Figure 3.1 Zero-Level DFD

2) First-level DFD:

- Shows how the system is divided into sub-systems, each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Online Blogging System as a whole.
- Identifies internal data stores of Web Page, Technology Blog, Comment, Blog Type, Create Blog that must be present in order for the online blogging system to do its job
- Shows the flow of data between various parts of Blog, Create Blog, Technology Blog, Web Page

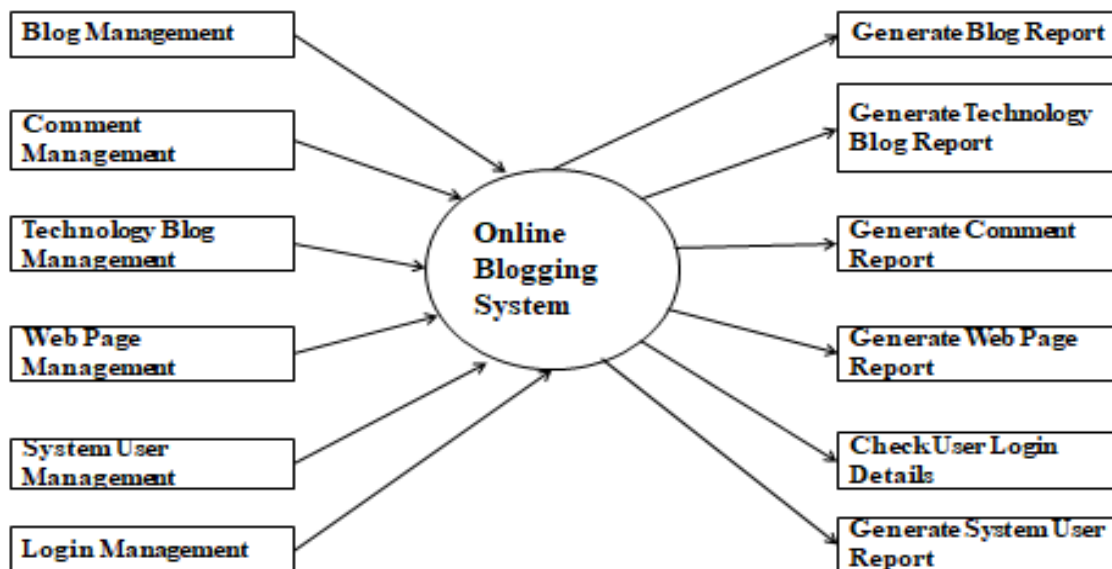


Figure 3.2 Level-One DFD

3)Second-level DFD:

- ✓ One step deeper into parts of level 1 of online blogging system
- ✓ Contains more details of Web Pages, Technology Blog, Comment, Blog Type, Create Blog, Blog Category and Blog
- ✓ Requires more functionality of online blogging to dive into the Second Level DFD

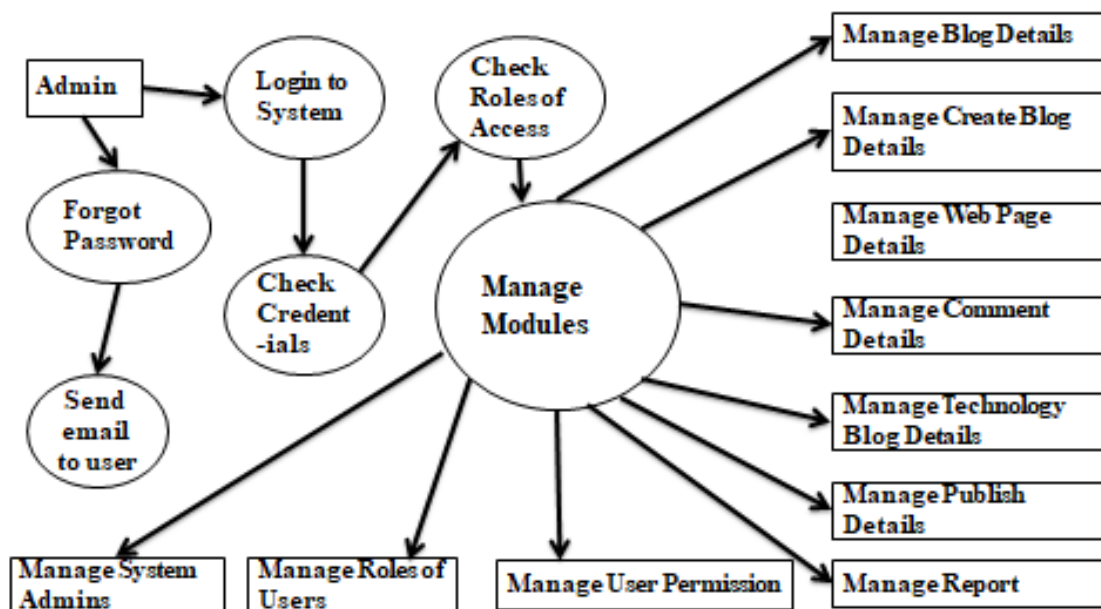


Figure 3.3 Level-Two DFD

Analysis Models

Sequence Diagrams

- ✓ UML sequence diagram of Blogging System which shows the interaction between the objects of Blog, Web Page, Blog Type, Category, and Comment.
- ✓ Demonstrates how the login page works in a blogging system.
- ✓ Instance of class objects involved in this UML Sequence Diagram of Blogging.System are as follows:-
 - Blog Object
 - Web Page Object
 - Blog Type Object
 - Category Object
 - Comment Object

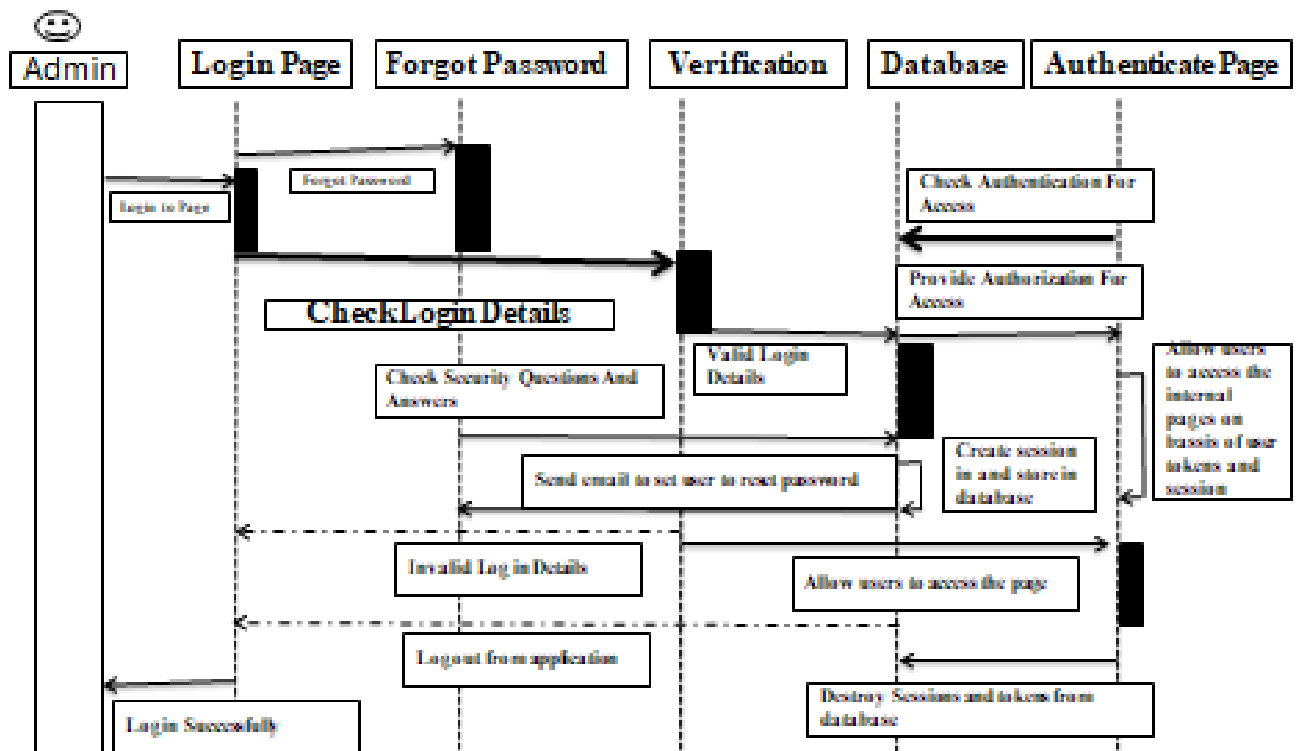


Figure 3.4 Sequence Diagram

Activity Diagram

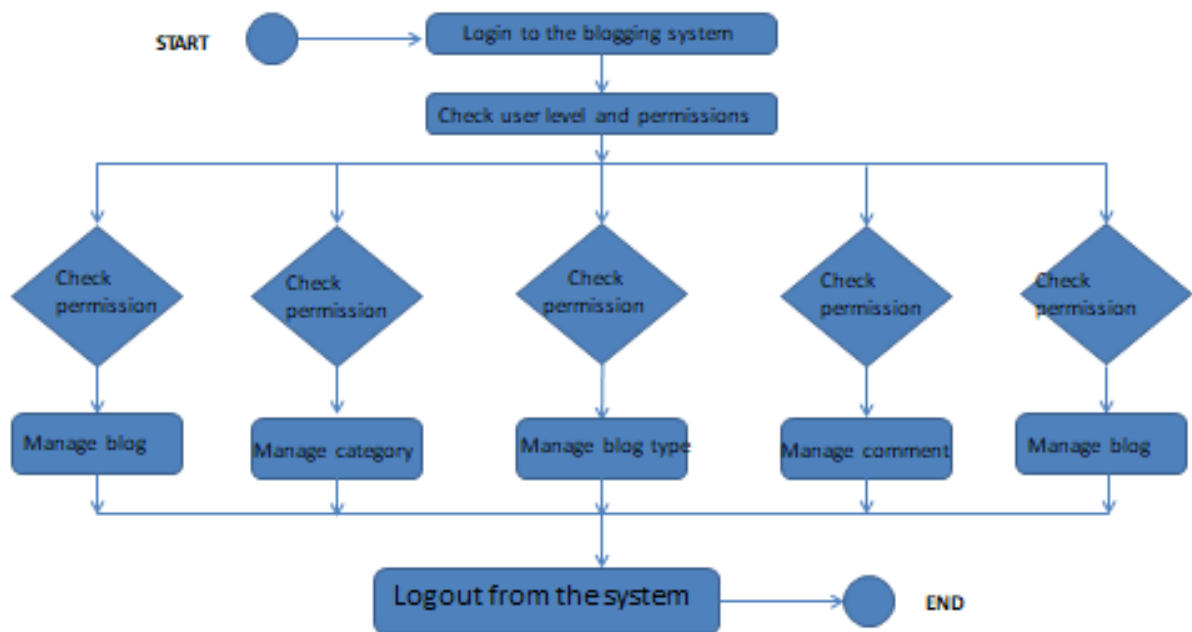


Figure 3.5 Activity Diagram

ER-Diagrams:**Blogging System entity and their attributes:****✓ Blog Entity:-**

blog_id, blog_user_id, blog_title, blog_type, blog_content,
blog_description

✓ Blog Category Entity:-

blog_category_id, blog_category, blog_category_title,
blog_category_type, blog_category_content,
blog_category_description

✓ Create Blog Entity:-

blog_id, blog_user_id, blog_title, blog_type, blog_content,
blog_description

✓ Comment Entity:-

comment_id, comment_user_id, comment_type, comment_title,
comment_description

✓ Blog Type Entity:-

blog_type_id, blog_type_name, blog_type_description

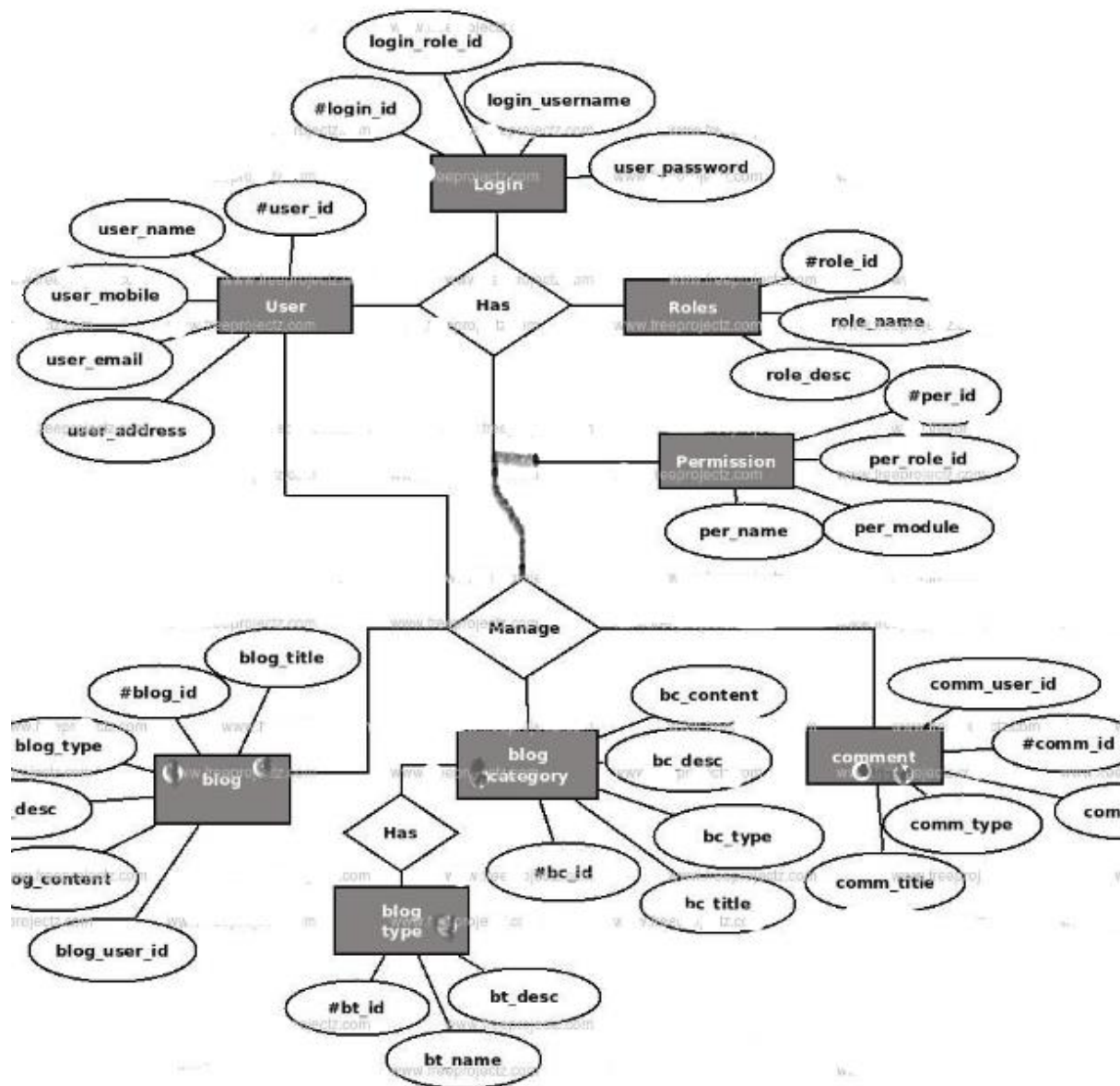


Figure 3.5 E-R Diagram

Chapter 4

Implementation And User Interface

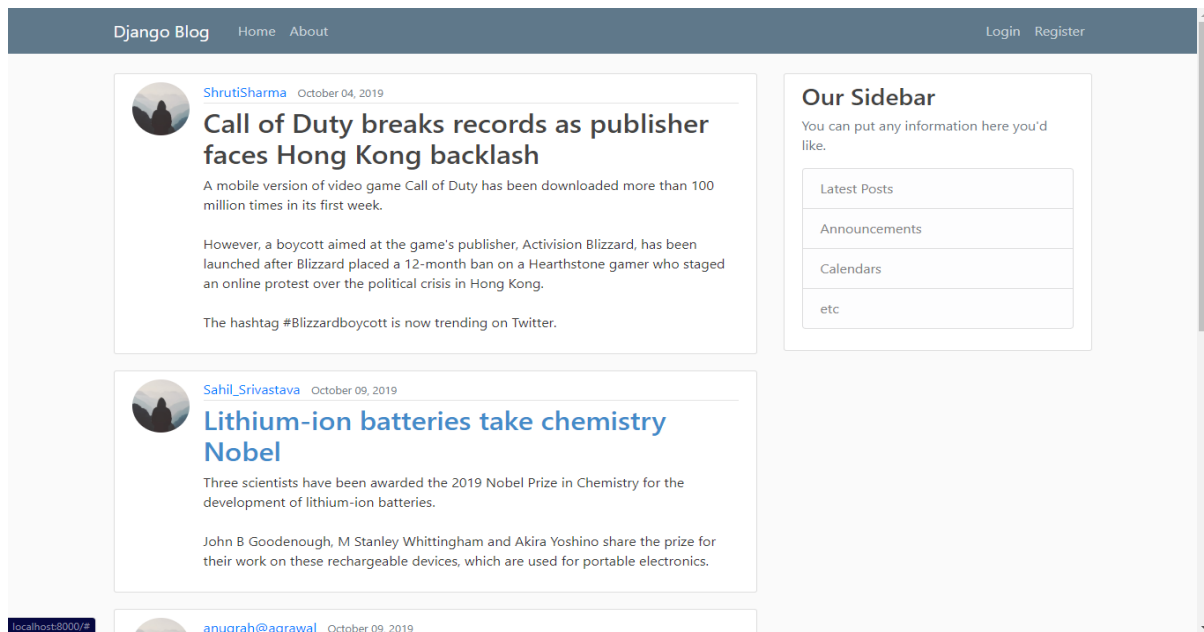


Figure 4.1 Blog Home Page

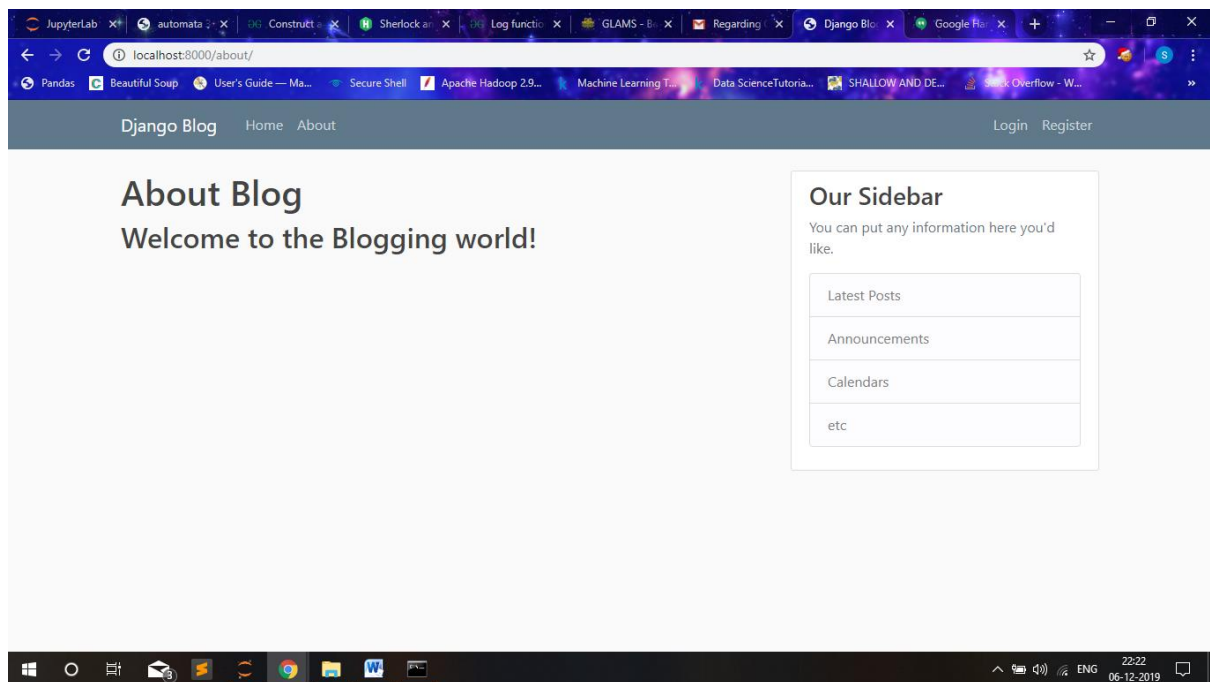


Figure 4.3 Blog About Page

ADMINS Tab:

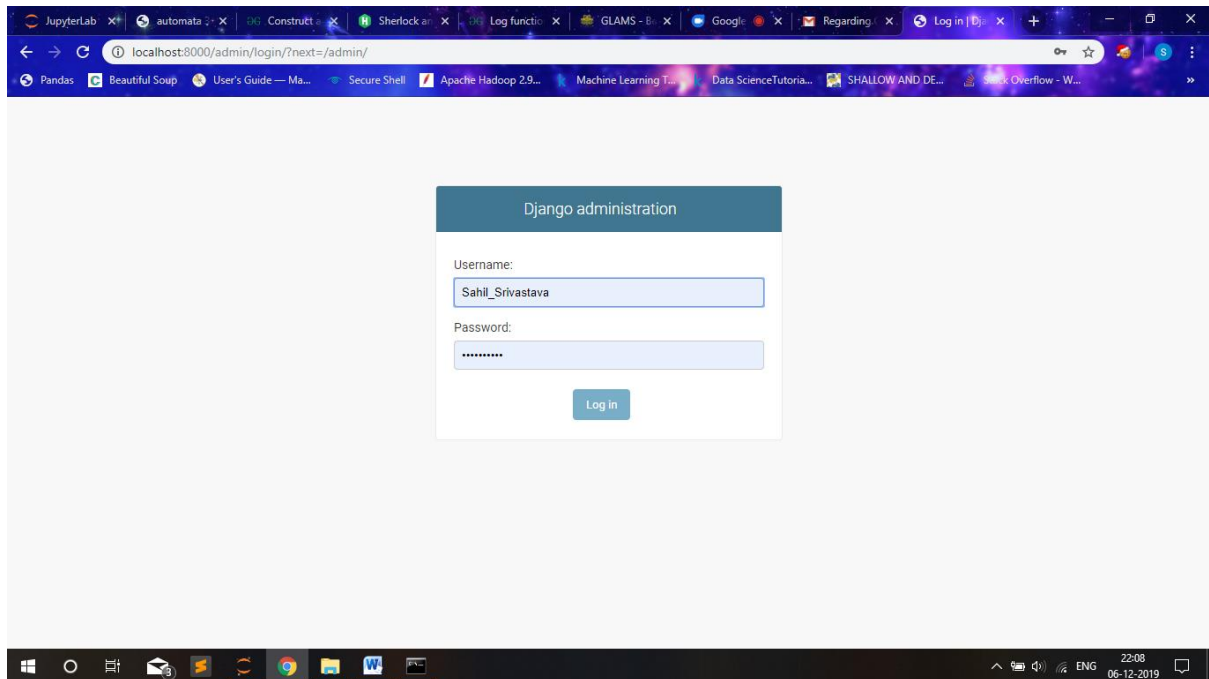
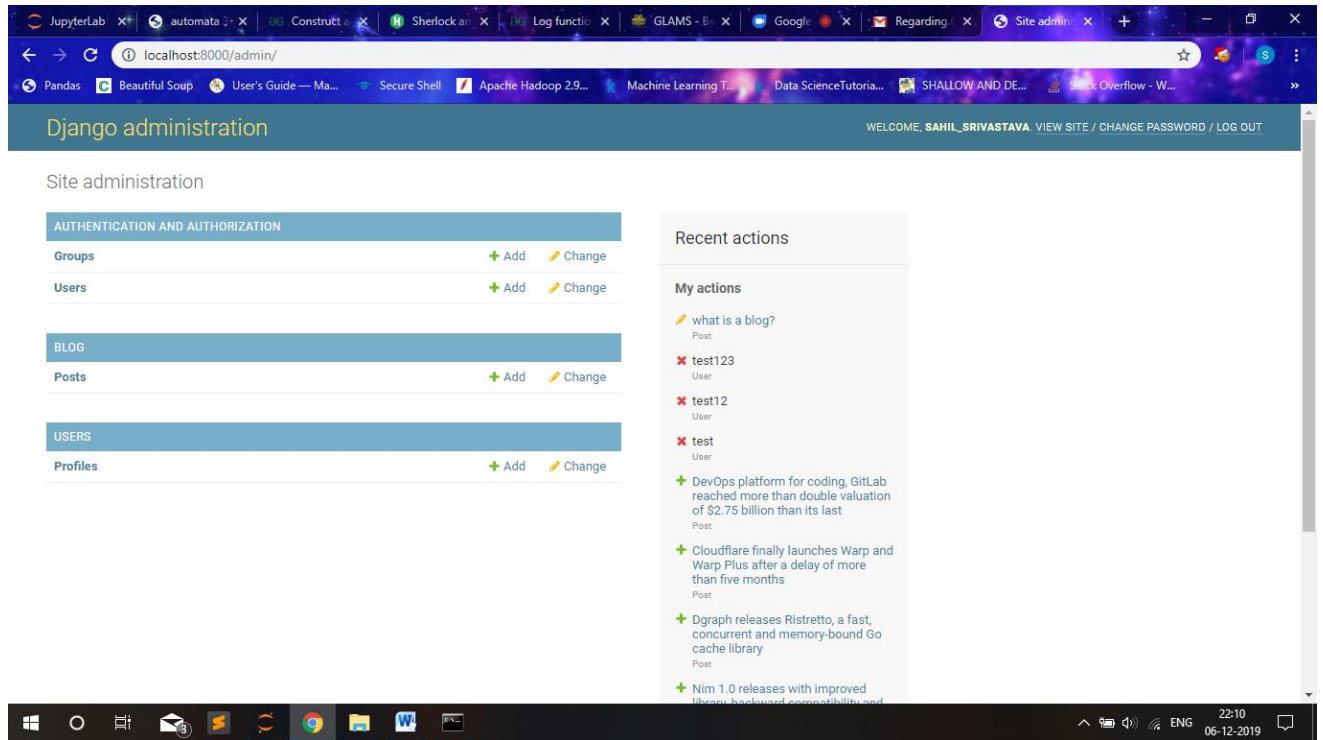


Figure 4.3 Admin Login Page

**Figure 4.4 Admin Panel Page**

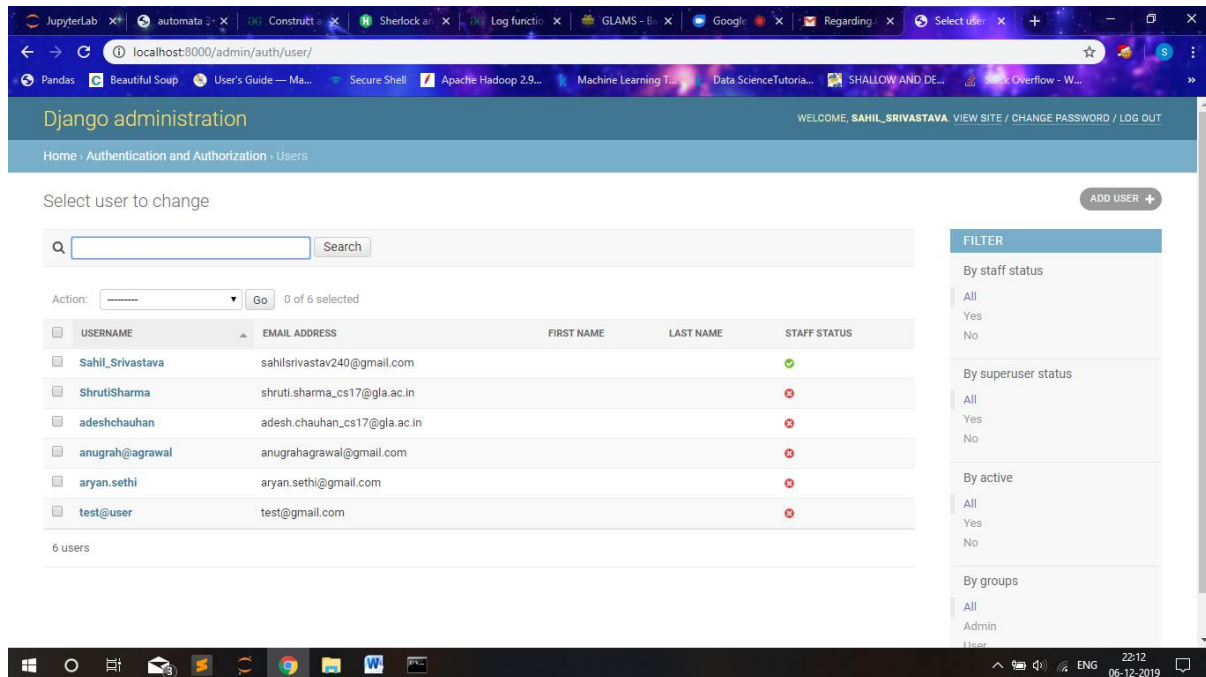
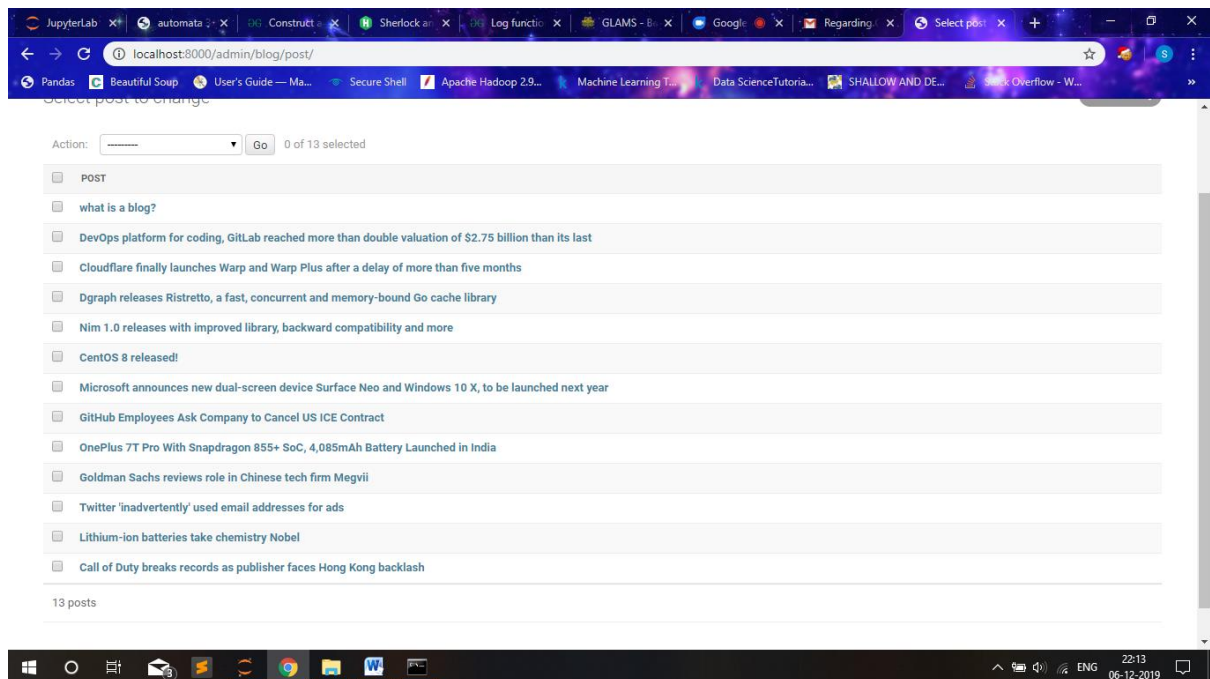


Figure 4.5 Admin User Category Page

**Figure 4.6 Admin Blogs Page**

Users Tab:

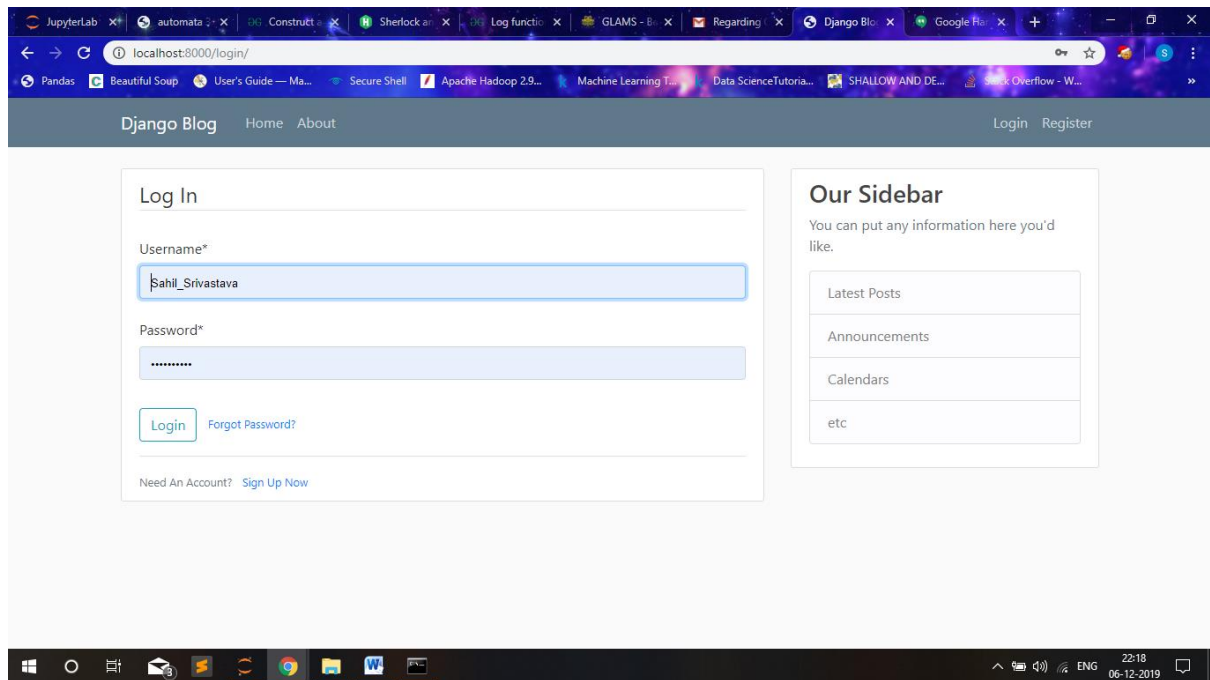


Figure 4.7 Login Page

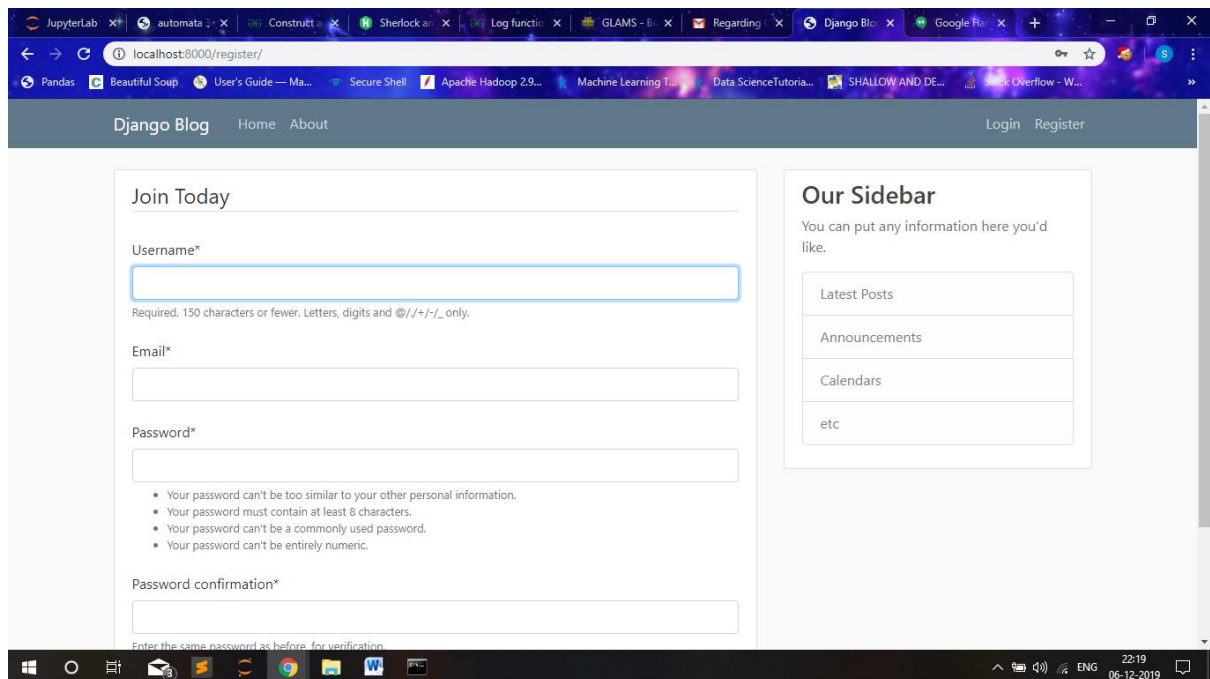


Figure 4.8 Registration Page

Chapter 5

Software Testing

5.1 Software Testing

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss.

5.2 Types of Testing

5.2.1 Unit Testing

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.

5.2.2 Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output.

5.2.3 Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

5.2.4 System Testing

In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.

In this we have security testing, recovery testing , stress testing and performance testing.

5.2.5 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

5.2.6 Black Box Testing

Black box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications.

In BlackBox Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

5.2.7 Alpha Testing

This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

5.2.8 Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment.

5.3 Test Cases

- 1) Username must be less than 150 characters
- 2) Password cannot be similar as the personal information
- 3) Password must contain at least 8 characters
- 4) Password cannot be entirely numeric
- 5) Password cannot be a commonly used password

Successfully Passed Test Cases

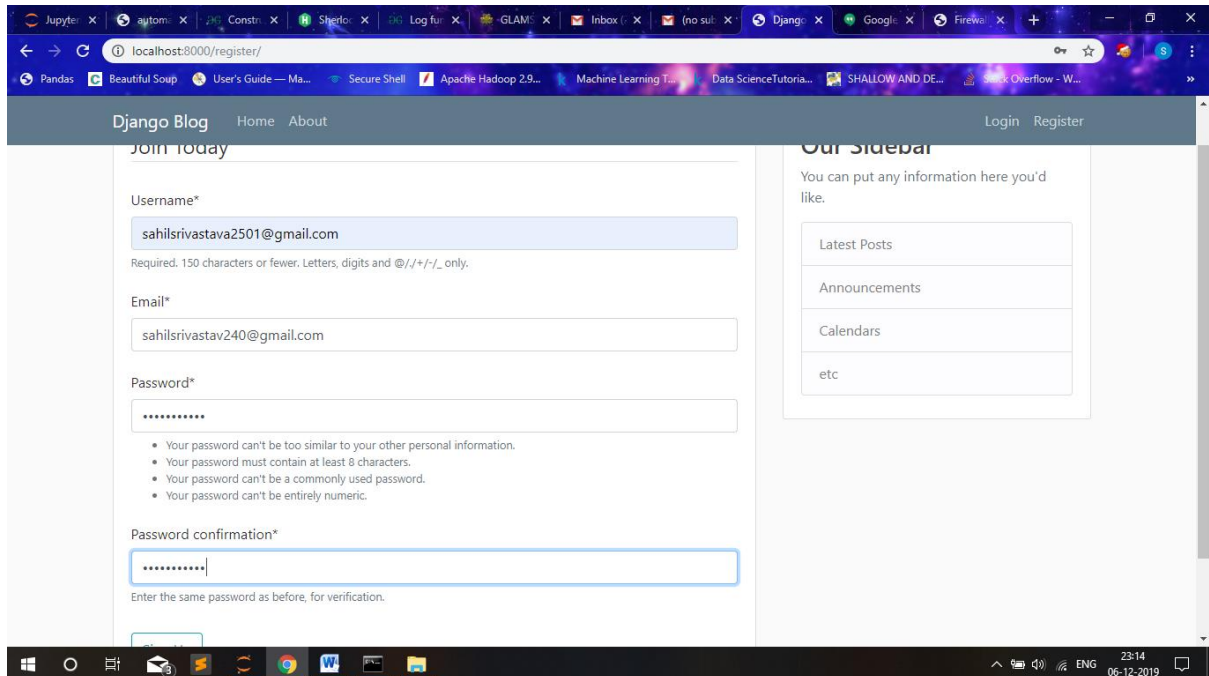


Figure 5.1

Failed Test Case

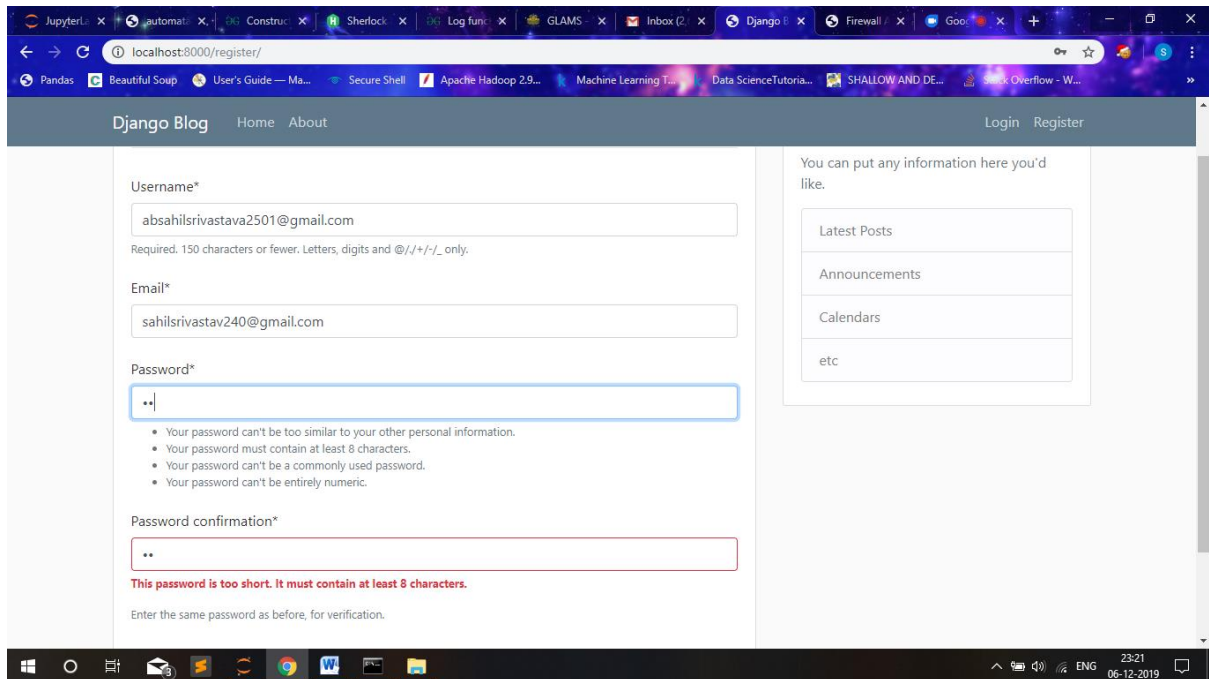


Figure 5.2

Chapter 6

...

FUTURE SCOPE OF THE PROJECT

Now-a-days, docker is being used widely across the globe. Docker enables you to rapidly deploy server environments in “containers.” Now, you might question why use Docker rather than VMware or Oracle’s VirtualBox?

While Docker utilizes the virtualization technology in the Linux kernel, it does *not* create virtual machines (in fact, if you run Docker on MacOS or Windows, you’ll have to run it on a virtual machine).

Instead of abstracting the hardware, containers abstract the OS. Each container technology features an explicit purpose, limiting the scope of the technology. Docker’s runs Linux, whereas Citrix’s XenApp runs Windows Server. Every container shares the exact same OS, reducing the overhead to the host system. Recall each VM runs its own copy of the OS, adding overhead for each instance. Containers exist to run a single application. Like XenApp, every Docker container targets a specific application. Docker also incorporates container management solutions for easy scripting and automation (especially important when considering containers’ focus on reducing execution time). A Docker container focuses on only one application at a time, and the environment is chromed to prevent access outside the container’s directory tree.

Chapter 7

Conclusion

Blogging is a great way to share your interests and ideas, educate, entertain people or promote your business. However, knowing where to start is not always easy. You need to pick your niche, type of blog to develop, a hosting company, writing platform... Sometimes, even coming up with the right colors of the website may seem to be difficult! A customer who having the account in the world can create a virtual account through this module. This module receives the customer profile details along with their password. Virtual account holders can login in to the system using this module. Thus this is the secured login page for the customers in the website. This module helps the user to reset their password for their account. This module is works for admin side mostly. The admin can update some account posts and delete account of some users for some offensive posts and comments after a certain period of warning. In this way, our blogging website is helpful.

References

1. <https://www.youtube.com/watch?v=90LCcim-wHQ>
2. <https://docs.docker.com>
3. <https://docs.djangoproject.com/en/3.0/>
4. [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))