**A Project Report**

**on**

**“Markov chain model of Rainfall Probability for Agricultural Planning in Anand”**



**Submitted By:**

**Rupesh N. Patel(06-0222-2016)**

**Major Guide: Minor Guide:**

Dr. R. S. Parmar EI. Vishal Mehra

Sign : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Certificate**

This is to certify that the project report entitled “Markov chain model of Rainfall Probability for Agricultural Planning in Anand”, in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Agricultural Information Technology** and submitted to collage of Agricultural information technology is an authentic record of my own work carried out under the supervision of Major Guide Dr. R.S. Parmar and Minor Guide Er. Vishal Mehra.

**Submitted** **By** **:**

Rupesh N. Patel

**Major Guide: Minor Guide:**

Dr. R. S. Parmar Er. Vishal Mehra

Sign : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgement**

**“Gratitude is a feeling which is more eloquent than words, more tranquil than silence…”**

The successful completion of a project is generally not an individual effort. It is an outcome of the collective efforts of several persons, each having its importance to the objective. This session is a vote of thanks to all those persons who have directly or indirectly added in their specials way towards the completion of this project.

I am heartily thankful to Dr. R. S. Parmar (Major Guide) and Er. Vishal Mehra (Minor Guide) for his time, suggestions, support and guidance at every step of this project and without whom, this project would have been a tough task.With this it would been a very difficult task without my classmates who were there at each time for the functioning and testing of this website. Thanks to all of them.

Thank you to Dr. D.R. Kathiriya (Principal and Dean, College of agricultural information technology, AAU, Anand) for providing this platform which helped in the development of my technical skills.

At last respect and thanks to all who directly or indirectly were part in the completion of this project.

**INDEX**

Contents

[1. Introduction : 5](#_Toc21741105)

[1.1 Objectives : 5](#_Toc21741106)

[1.2 Scope : 5](#_Toc21741107)

[2. System Requirements : 5](#_Toc21741108)

[2.2.Software Configuration : 5](#_Toc21741109)

[2.2.Hardware Configuration(Current): 6](#_Toc21741110)

[2.3.Hardware Configuration(Required Minimum): 6](#_Toc21741111)

[3. Literature Review 6](#_Toc21741112)

[3.1. Django : 6](#_Toc21741113)

[3.2. Working of Django : 6](#_Toc21741114)

[3.3 The Django Framework : 7](#_Toc21741115)

[3.4. Advantages of Django Framework : 7](#_Toc21741116)

[3.5. Dis-advantages of Django Framework : 8](#_Toc21741117)

[3.6. Programming language 8](#_Toc21741118)

[3.7. Server and Client Technology 8](#_Toc21741119)

[3.8. Development Environment 8](#_Toc21741120)

[3.9. Description of tools used 9](#_Toc21741121)

[3.10. Features of Visual Studio Code 9](#_Toc21741122)

[3.11. Desining Documentation 11](#_Toc21741123)

[3.12 JavaScript Documentation 11](#_Toc21741125)

[4.Module List: 11](#_Toc21741126)

[4.1. Complaint Module 11](#_Toc21741127)

[4.2. Product Sellig 11](#_Toc21741128)

[4.3 Camera and Server Module 12](#_Toc21741129)

[5.Testing: 12](#_Toc21741130)

[5.1 Unit Testing 12](#_Toc21741131)

[5.2.Integration Testing 13](#_Toc21741132)

[5.3.System Testing 14](#_Toc21741133)

[5.4.System Integration Testing 14](#_Toc21741134)

[6.Data Dictionary: 15](#_Toc21741135)

[6.1. Product Table 15](#_Toc21741136)

[6.2.User Registration Table 16](#_Toc21741197)

[6.3.Engineer Registration Table 16](#_Toc21741243)

[6.4.Admin or Head Table 17](#_Toc21741294)

[6.5.Transection Table 17](#_Toc21741325)

[6.6.Complaint Table 18](#_Toc21741391)

[6.7.Server Table 19](#_Toc21741477)

[6.8.Camera Table 19](#_Toc21741523)

[6.9.Video Table 20](#_Toc21741564)

[6.10.Cart Table 20](#_Toc21741595)

[6.11.Order Table 21](#_Toc21741636)

[7.System Diagrams 22](#_Toc21741707)

[7.1. Context Diagram or Level-0 Diagram 22](#_Toc21741708)

[7.2. Entity Relationship Diagram 23](#_Toc21741709)

[7.3. Level-1 Diagram 24](#_Toc21741710)

[8.System Screenshots 25](#_Toc21741711)

[8.1. General Screenshots 25](#_Toc21741712)

[8.2. Customer Login 28](#_Toc21741713)

[8.3. Engineer Login 30](#_Toc21741714)

[8.4. Admin Login 31](#_Toc21741715)

[9. Gannt Chart: 35](#_Toc21741716)

[10. Conclusion: 35](#_Toc21741717)

[11. References: 36](#_Toc21741718)

# 1. Introduction :

Markov chain model of the Rainfall Probability system provides an easy interface between a user and the Marko chain model. This model used for the yield of crop particularly under rain fed conditions depends on the rainfall pattern. Simple criteria related to sequential phenomena like dry and wet spells could be used for analyzing rainfall data to obtain specific information recognized as a suitable model to explain the long term frequency behavior of wet or dry spells. Several authors have demonstrated its practical utility in agricultural planning for both long and short term periods. This model enables us to determine the probability of occurrence of dry and wet spells during a particular week and also finding the Descriptive Analysis.

## Objectives :

* To create DBMS for Rainfall data of Agro meteorological observatory of Anand.
* To develop tool for Markov Model for wet and dry analysis, Descriptive data analysis and Rainfall data Normal.
* To harness Information Technology to achieve the above objectives.

## Scope :

Web Based Markov Model for wet and dry analysis, tools is the web based system using which Scientist can generate graphs and reports based on the weather data by the observatory for various purpose of their research or Agricultural Planning.

# 2. System Requirements :

## 2.2.Software Configuration :

* Tools or Text Editor : Vs code or Jupyter notebook, SQLite studio
* Browser : Any Latest Updated Browser
* Language: Python
* Database: Sqlit3

## 2.2.Hardware Configuration(Current):

* Processor : Intel core I3
* RAM : 8 GB
* Hard Disk : 1 TB

## 2.3.Hardware Configuration(Required Minimum):

* Processors: Intel Atom® processor or Intel® Core™ i3 or higher
* RAM: 1 GB
* Hard Disk : As per User Requirement.

# 3. Literature Review

## 3.1. Django :

* **Django** was created in the fall of 2003, when the web programmers at the Lawrence Journal-World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications. It was released publicly under a BSD license in July 2005. The framework was named after guitarist **Django** Reinhardt.

## 3.2. Working of Django :

* Django can look complicated at first. Just navigating to a single, simple page can involve code in three or four different files. Django utilizes a design paradigm widely known as "MVC architecture", or rather, a slight

variation they tend to refer to as "MTV". In this the code is split between a Model, a View, and a Controller. Or, in Django's version, a Model, Template, and View. While this isn't going to attempt to be a thorough explanation of this system ([The Django Book's first chapter](http://www.djangobook.com/en/2.0/chapter01/) already does a much better job of that than I could ever attempt) it will attempt to walk through a section of our code and explain how it works and how to trace your way through the process that renders a webpage when you navigate to a URL.

## 3.3 The Django Framework :

* Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

## 3.4. Advantages of Django Framework :

* + **Fast :**This has been designed in a way to help the developers make an application as fast as possible. From idea, production to release, Django helps in making it both cost effective and efficient. Thus it becomes an ideal solution for developers having a primary focus on deadlines.
  + **Fully Loaded :**It works in a way that includes dozens of extras to help with user authentication, site maps, content administration, RSS feeds and much more such things. These aspects help in carrying out the web development process completely.
  + **Secure :** When you are doing it in Django, it is ensured that developers don’t commit any mistakes related to security. Some of the common mistakes include SQL injection, cross-site request forgery, clickjacking and cross-site scripting. To manage effectively usernames and passwords, the user authentication system is the key.
  + **Scalable :** To meet the heaviest traffic demand, the benefits of Django framework can be seen. Therefore, the busiest sites use this medium to quickly meet the traffic demands.

## 3.5. Dis-advantages of Django Framework :

* Uses routing pattern specify its URL.
* Django is too monolithic.
* Everything is based on Django ORM.
* Components get deployed together.
* Knowledge of full system is required to work*.*

## 3.6. Programming language

* Python

## 3.7. Server and Client Technology

* Wsgi(Web Server Gateway Interface) server.
* Also run on Apache, Gunicorn, Nginx servers.

## 3.8. Development Environment

* Visual Studio Code(IDE) in my Project.
* Support any text editor.

## 3.9. Description of tools used

* Working with Python in Visual Studio Code, using the Microsoft Python extension, is simple, fun, and productive.
* The extension makes VS Code an excellent IDE, and works on any operating system with a variety of Python interpreters.
* It leverages all of VS Code's power to provide auto complete and IntelliSense, linting, debugging, and unit testing, along with the ability to easily switch between Python environments, including virtual and conda environments.

## 3.10. Features of Visual Studio Code

#### 1. Intellisense

IntelliSense is code **auto-complete meets artificial intelligence**. This utility provides a list of suggestions along with a short hint or description as we are writing codes. These is **derived from several contextual factors** such as the programming language, the syntax, the variables, the functions as well as all the codes within the file.

IntelliSense supports a number of programming language out-of-the-box, including Sass, LESS, JavaScript, TypeScript, and PHP. Some extensions to include IntelliSense for other programming languages are also available. It is a great feature to improve productivity.

#### 2. Peek

While coding, you may often forget a particular function, where the function is initially defined, and what are the required parameters.

With **Peek**, you can select a function then hit Shift + F12. The selection **expands into an inline window showing the complete definition of the function** as well as where the function is defined. The feature currently works in C, C#, JavaScript, TypeScript, .NET, and a few other programming languages.

#### 3. CLI

Similar to SublimeText with its subl command line, Visual Studio Code is equipped a CLI named code and it is easy to install.

In Windows, the CLI will have already been installed along with the app. The CLI is accessible through the Command Prompt. In OS X, it can be installed and uninstalled through the Command Palette.

You can use the CLI to open a particular directory to Visual Studio Code, or open your current project directory in Terminal or Command Prompt directly from the editor.

#### 4. Built-in Git

Visual Studio Code has Git built right in the editor. On the left of the sidebar, you will find the Git icon where you can [initialize Git](https://git-scm.com/docs/git-init) as well as **perform several Git commands** such as commit, pull, push, rebase, publish, and look into the changes within the file.

In addition, if you are making changes on a Git repository, the Visual Studio will **show color indicators in the code editor gutter**, indicating where you have made the modifications.

#### 5. Task Runner

Last but not least, Visual Studio Code also has a built-in Task Runner, which provides some level of convenience.

To use this feature, we can either set a tasks.json file or through some popular Task configuration like Grunt, Gulp, or MSBuild if provided. Once the configuration is set, we can run the Task Runner through the Command Palette by typing **Run Task**. Typing **Tasks** will list all related commands.

## 3.11. Desining Documentation

## Twitter Bootstrap is the most popular front end frameworks currently. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and JavaScript.

* CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

## 3.12 JavaScript Documentation

* JavaScript is a programming language that lets you supercharge your HTML with animation, interactivity, and dynamic visual effects. JavaScript can make web pages more useful by supplying immediate feedback. For example, a JavaScript-powered automatic calculation can instantly display a total cost, with tax and estimated cost. JavaScript can produce an error message immediately after someone attempts to submit a web form that’s missing necessary information.

# 4.Module List:

## 4.1. Complaint Module

* In This Module Customers Send their complaint and engineer can accept customers complaint.
* Engineer can sent bill for his services.
* Customer can give feedback for engineer.

## 4.2. Product Sellig

* This module admin can upload a products(Computer Related Products).
* Customer can add product in cart and buy a product and give rating for the products.

## 4.3 Camera and Server Module

* This module admin can upload camera and accept a contract for the camera.
* Also admin can accept server made order as per customer request.

# 5.Testing:

### 5.1 Unit Testing

#### 1.Objective

* The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Unit testing tests the minimal software component, or module.
* Each unit of the software is tested to verify that the detailed design for the unit has been correctly implemented. In an object-oriented environment this is usually at the class level, and the minimal tests include the constructors and the destructors.
* Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews

#### 2.Input

* Unit Test Specifications
* Code to be tested

#### 3.Testing Process

* Checking for availability of Code Walk-through reports which have documented the existence of and conformance to coding standards.
* Review of Unit Test Specifications.
* Verify the Unit Test Specifications conform to the program specifications.
* Verify that all boundary and null data conditions are included.

**4.Following are some of the test cases that are given below:**

* Registration Form Validations Testings.
  + Text Field.
  + Number Field.
  + Email Field.
  + Password Field.
  + Regular Expression Validation for each Fields.
* Module Testing
  + Complaint.
  + Product Selling.
  + Camera and Server Contracts.

### 5.2.Integration Testing

* Integration testing exposes defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of architectural design are integrated and tested until the software works as a system.
* As Modular coding strategy was used, after completion of my module and integrating the module with the complete application, time was given to me to test their part of module completely and thoroughly.
* Integration testing takes as its input, modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

**Following are some of the test cases that are given below:**

1. Verify that communication between the systems are done correctly

2. Understand how the data is transferred from one module to another.

3. Is record fetch automatically from one module to another?

### 5.3.System Testing

* System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.
* System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

### 5.4.System Integration Testing

* System [Integration Testing](https://www.guru99.com/integration-testing.html) is defined as a type of software testing carried out in an integrated hardware and software environment to verify the behavior of the complete system. It is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirement.
* System Integration Testing (SIT) is performed to verify the interactions between the modules of a software system. It deals with the verification of the high and low-level software requirements specified in the Software Requirements Specification/Data and the Software Design Document.
* It also verifies a software system's coexistence with others and tests the interface between modules of the software application. In this type of testing, modules are first tested individually and then combined to make a system.
* For Example, software and/or hardware components are combined and tested progressively until the entire system has been integrated

# 6.Data Dictionary:

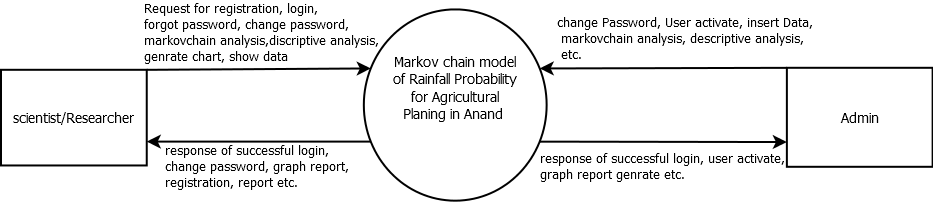
## 6.1. Product Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Element | Alias | Data Type | Nullable | Constrain | Description |
| Product Id | p\_id | Int | Not Null | PK | The unique identification key for product |
| Product Name | p\_name | Varchar(20) | NotNull | - | The field for product name |
| Product Price | price | int | NotNull | - | The field for product price in Rs./- |
| Product Company | company | Varchar(20) | NotNull | - | The field for product company |
| Product Stock | stock | Int | NotNull | - | The field for product stock available |
| Product Rating | rating | Float(15) | NotNull | - | Product rating field for customers review. |
| Rating Count | crating | Int | NotNull | - | Field for product rating count of customer |
| Product Description | desc | Varchar(200) | NotNull | - | Field for description of the product |
| Product Warranty | warranty | Float(10) | NotNull | - | Field for product warranty in years |
| Product Image | image | Image | NotNull | - | Field for product image |
| Product Date | pdate | datetime | NotNull | - | Field for product launch date |

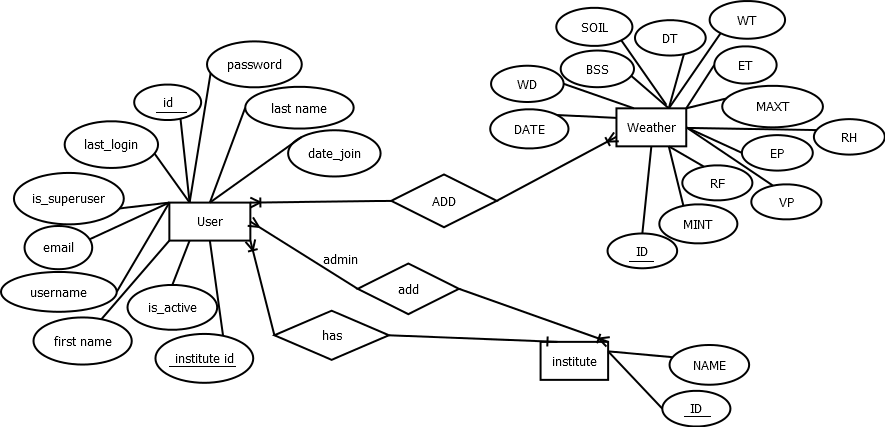
# 

# 7.System Diagrams

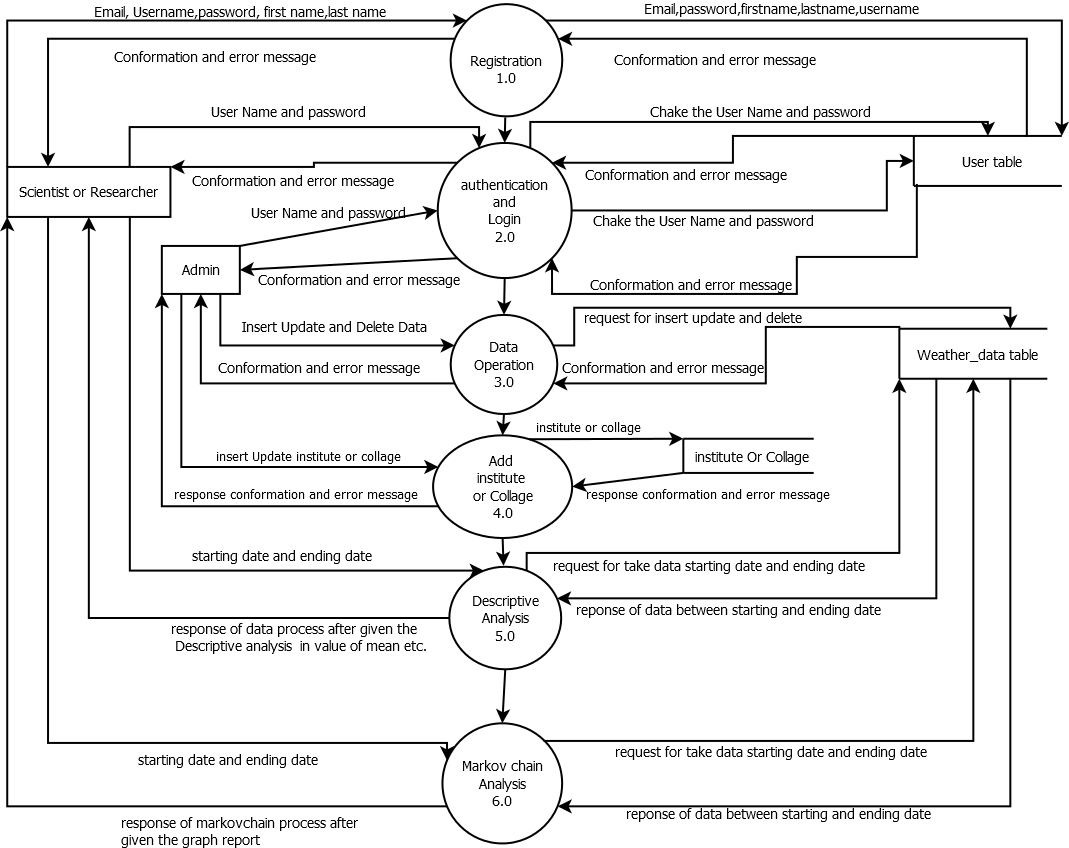
## 7.1. Context Diagram or Level-0 Diagram



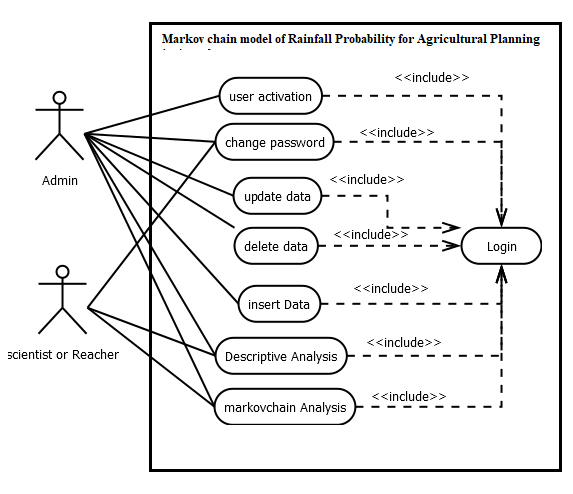
## 7.2. Entity Relationship Diagram



## 7.3. Level-1 Diagram



## 7.4 Use case Diagram



## 7.5 Class Diagram

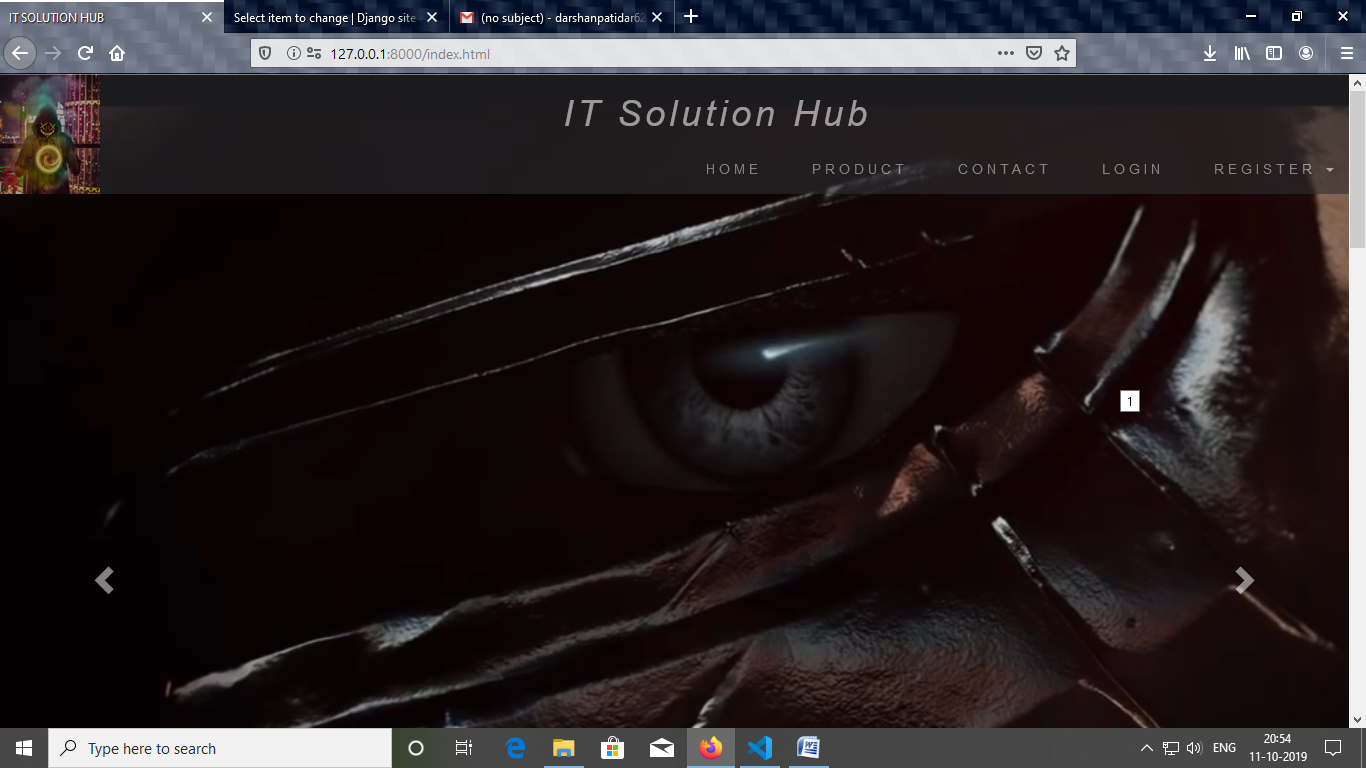
## 7.6 Activity Diagram

# 8.System Screenshots

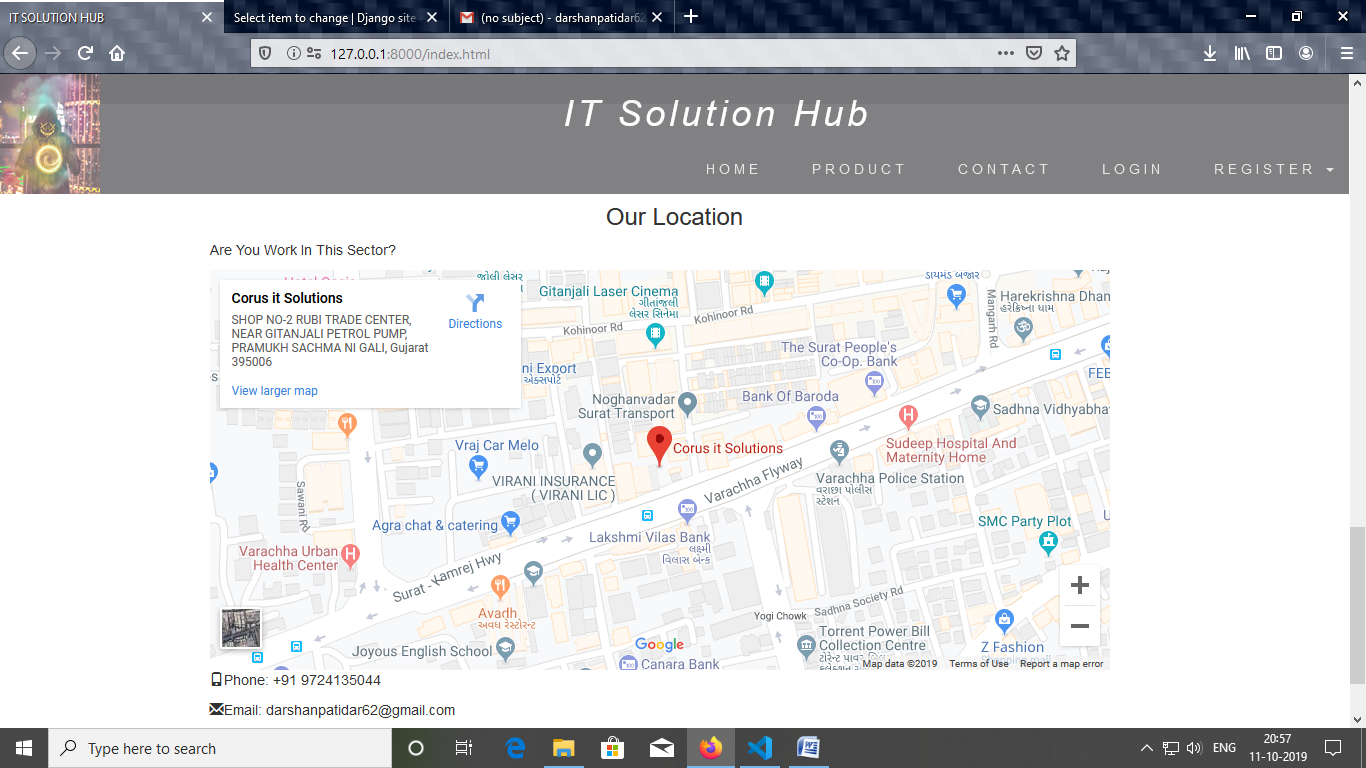
## 

## 8.1. General Screenshots

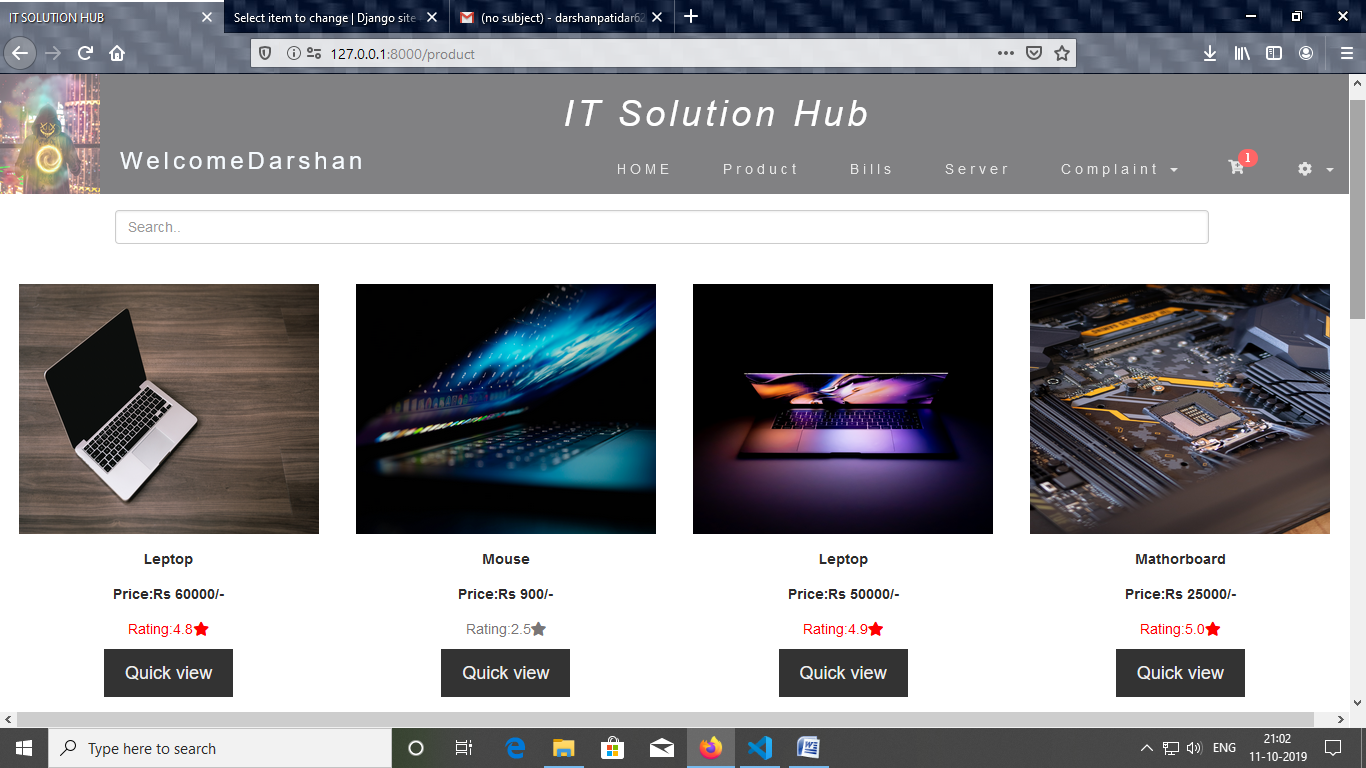
1.Home Page



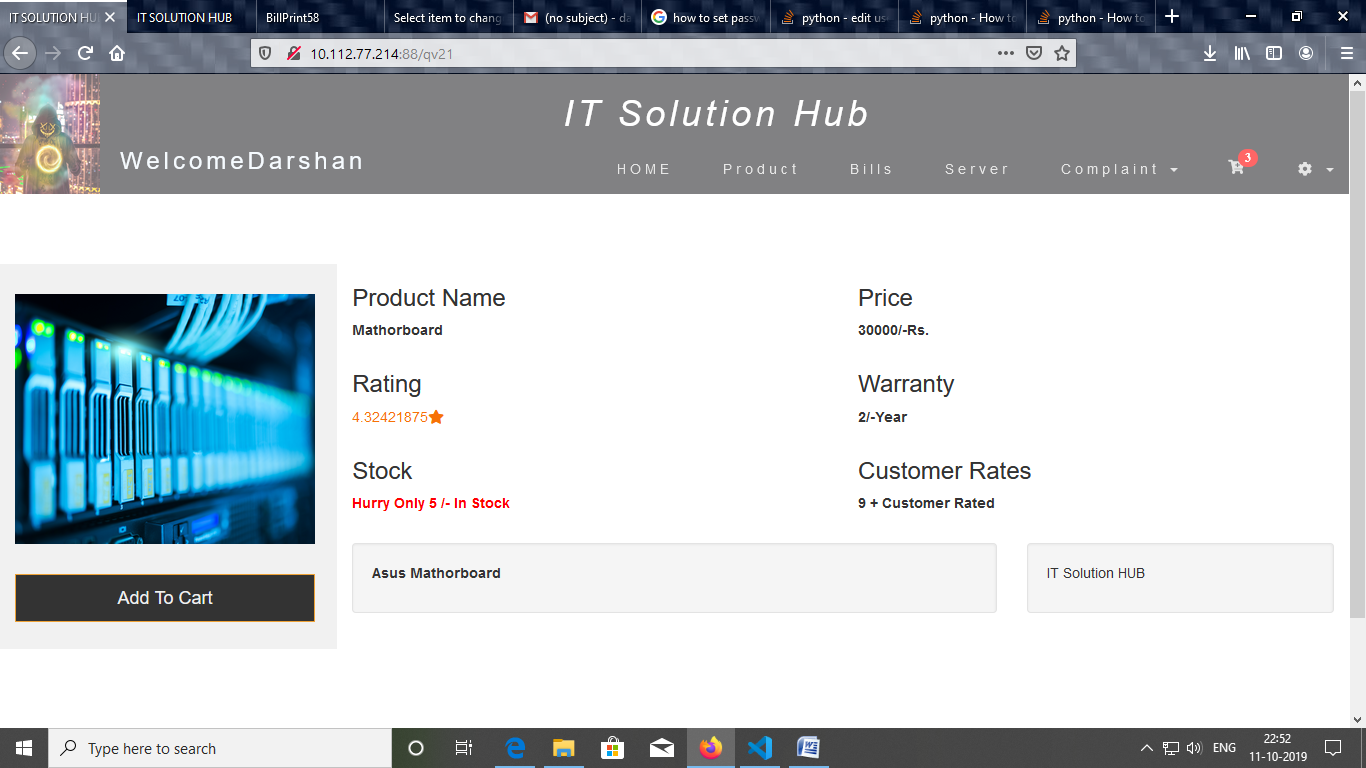




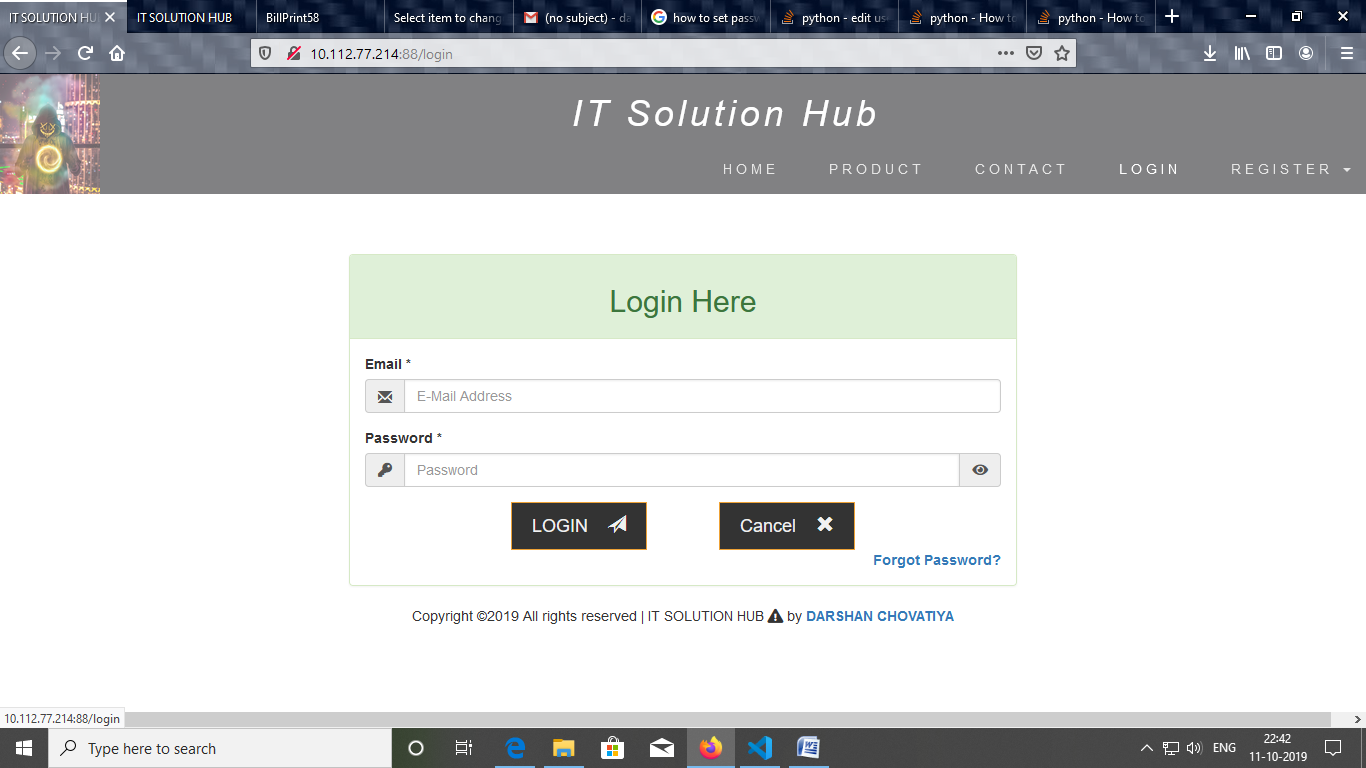
2.Product Page



3.Quick View Page



4.Login Page



4.Registration Page

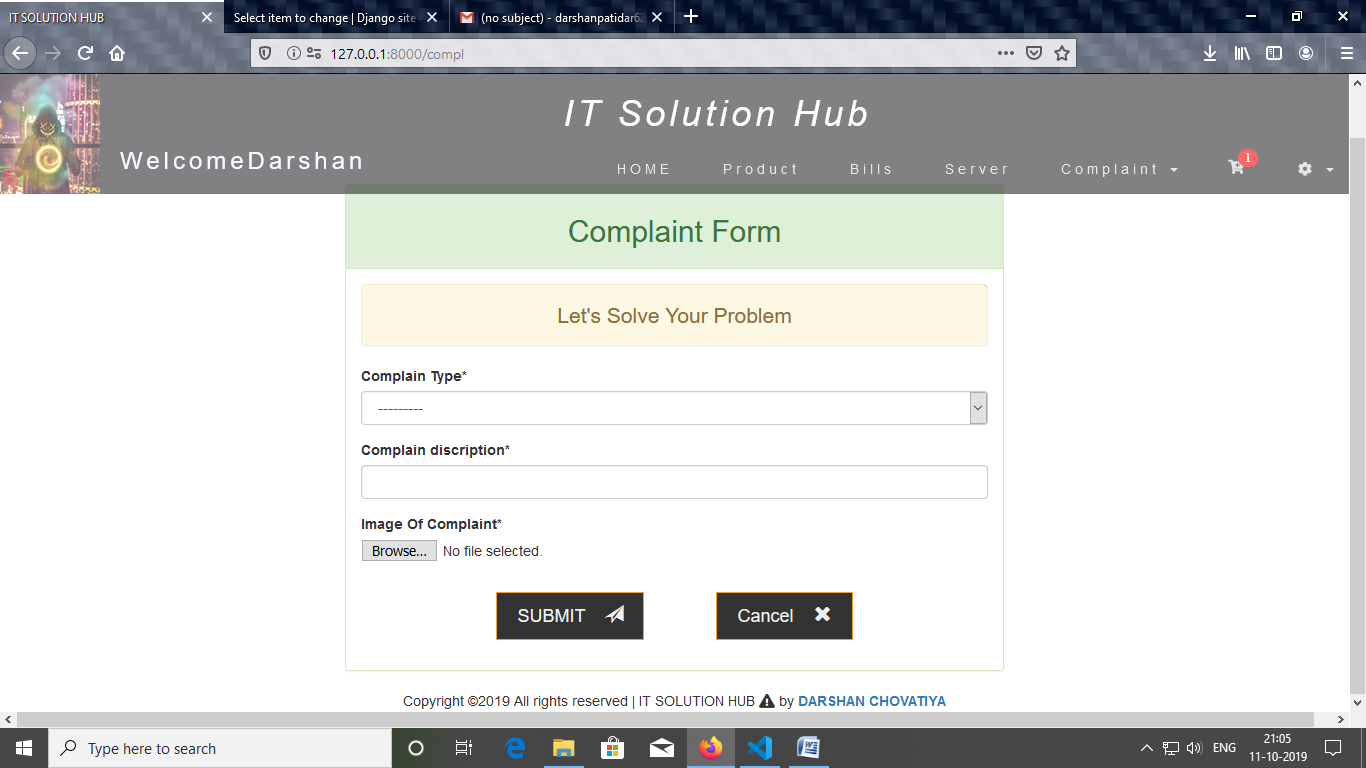
# Screenshot (62).png

6.Edit Profile Page

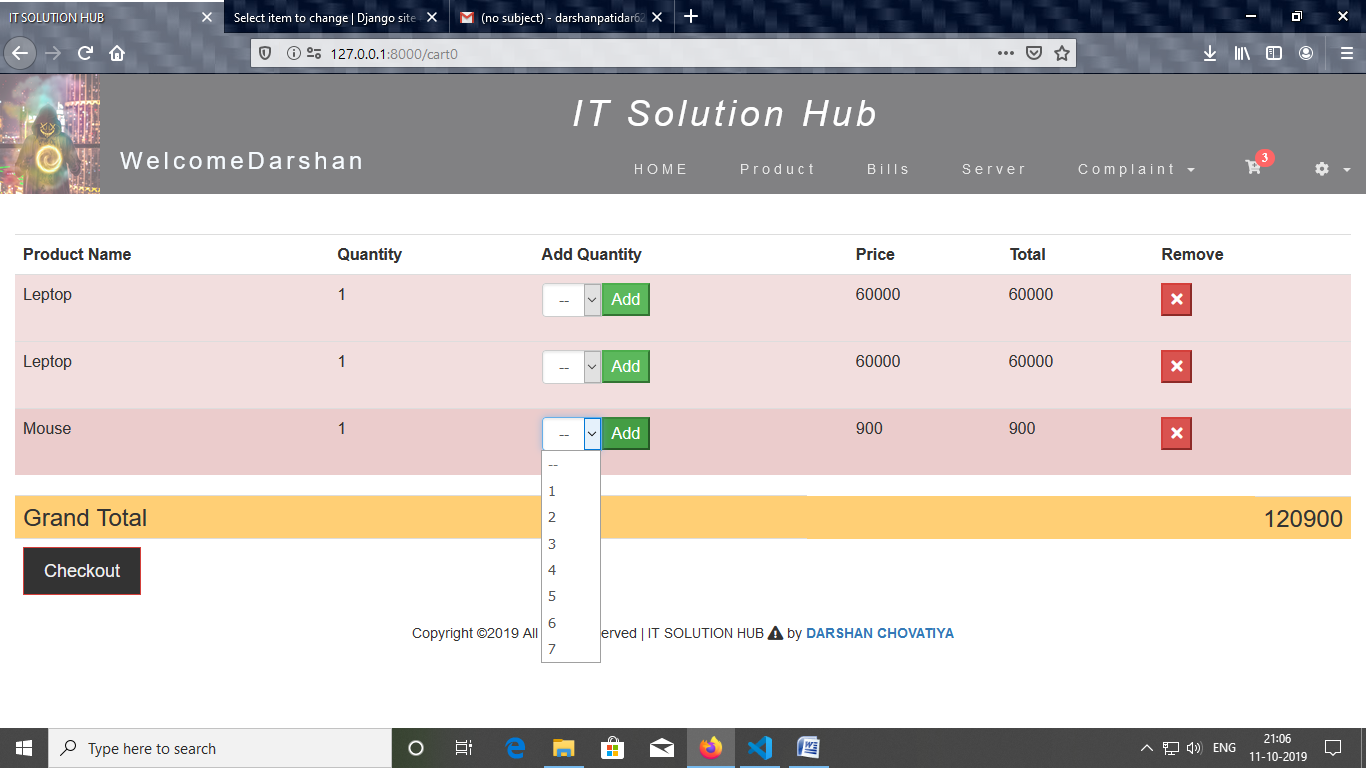
****

## 8.2. Customer Login

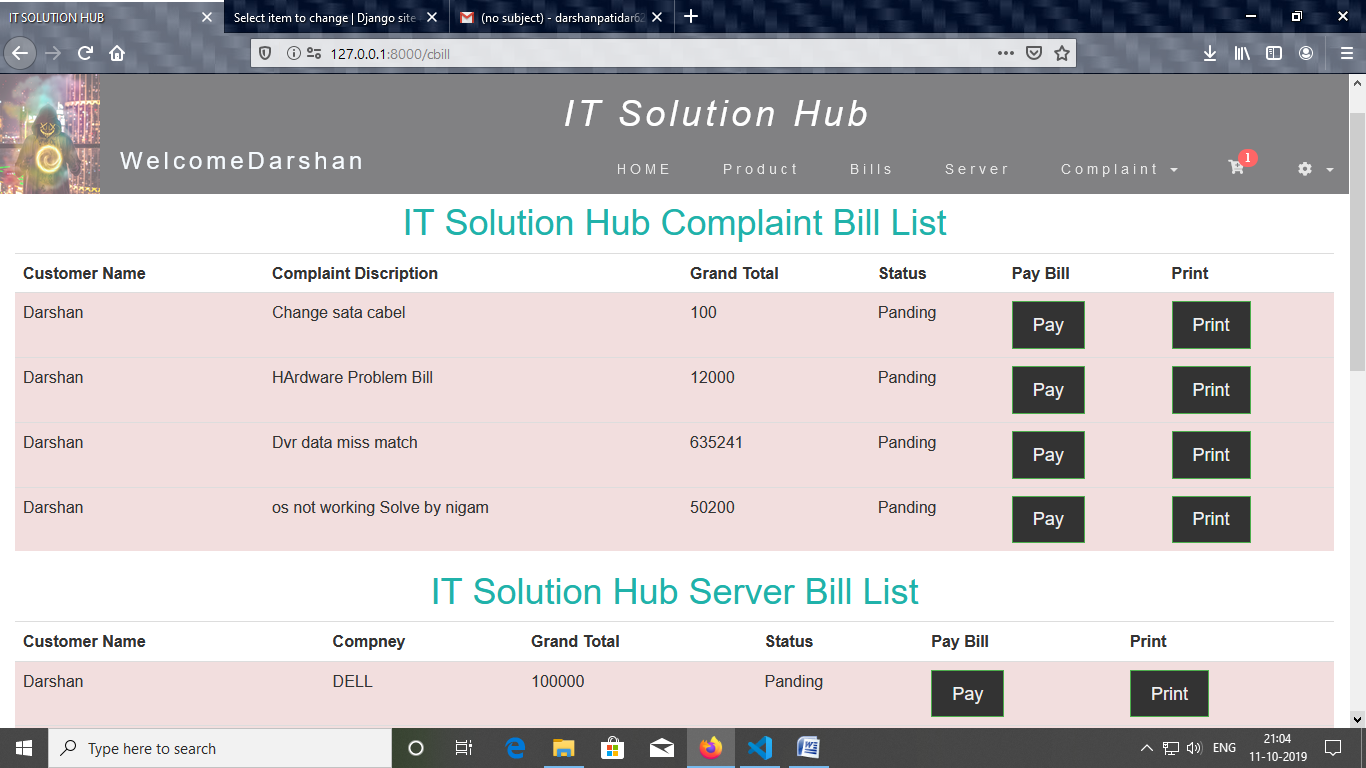
1.Complaint Page



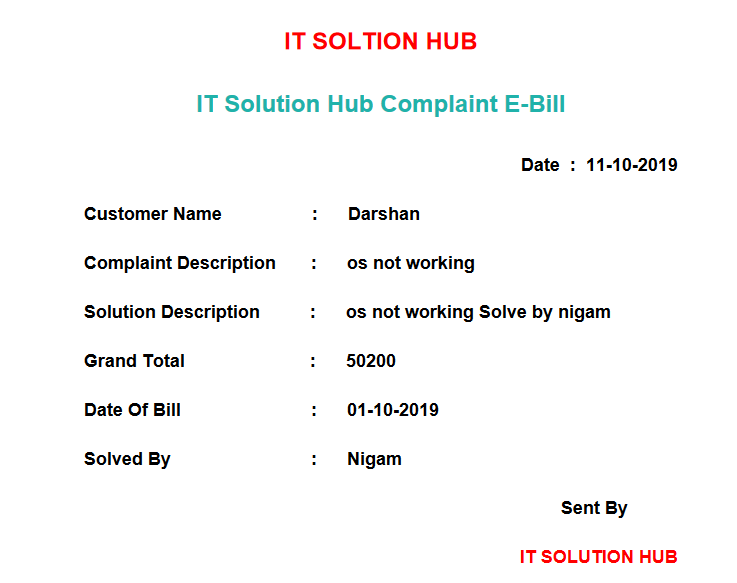
2.Cart Page



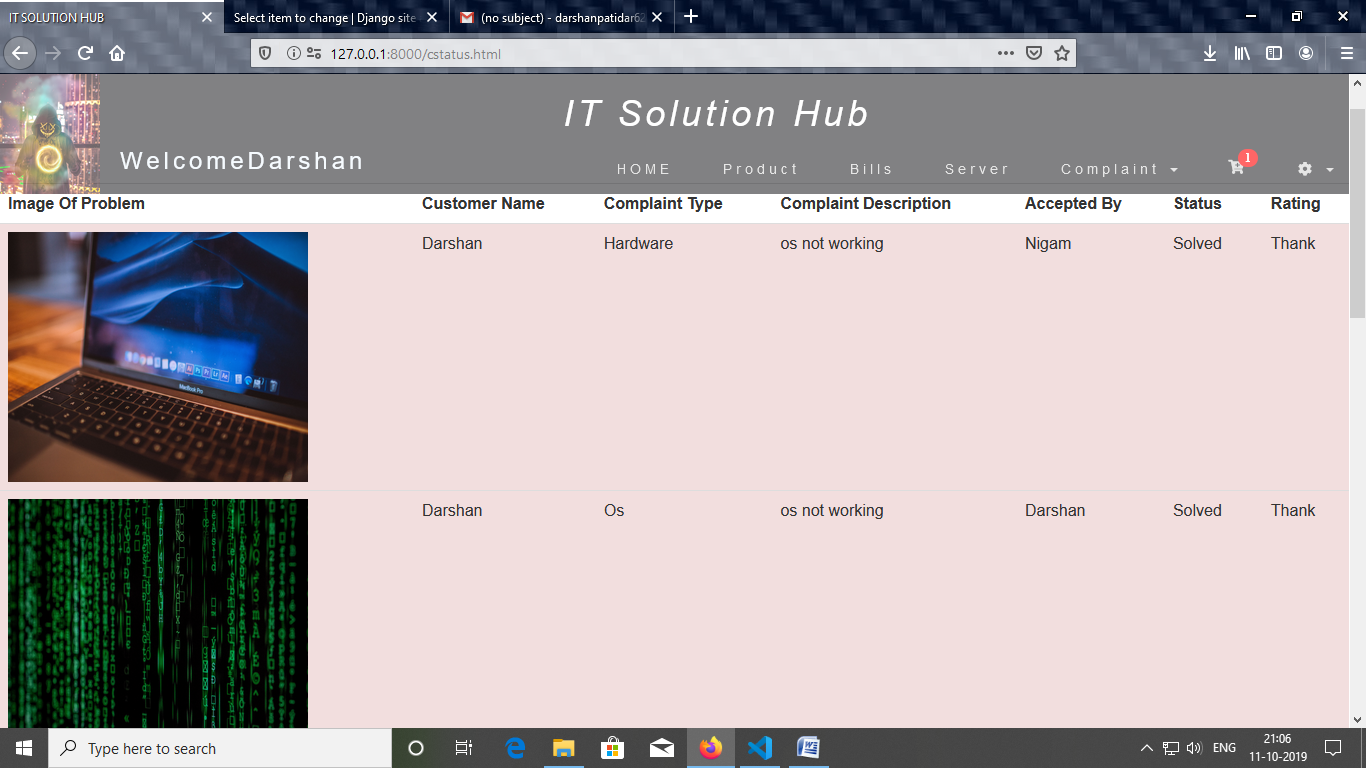
3.Complaint Bill Page



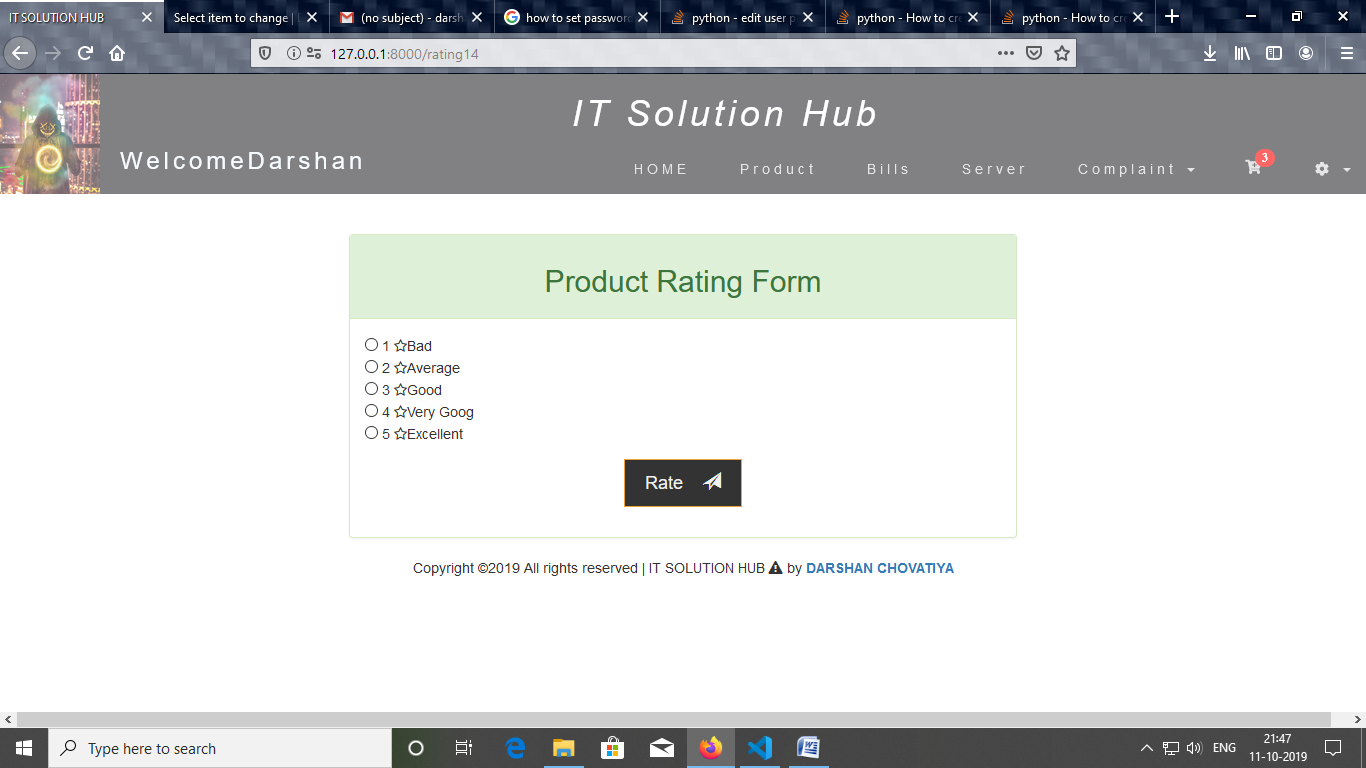
4. Bill Print PDF Page



5.Complaint Status Page

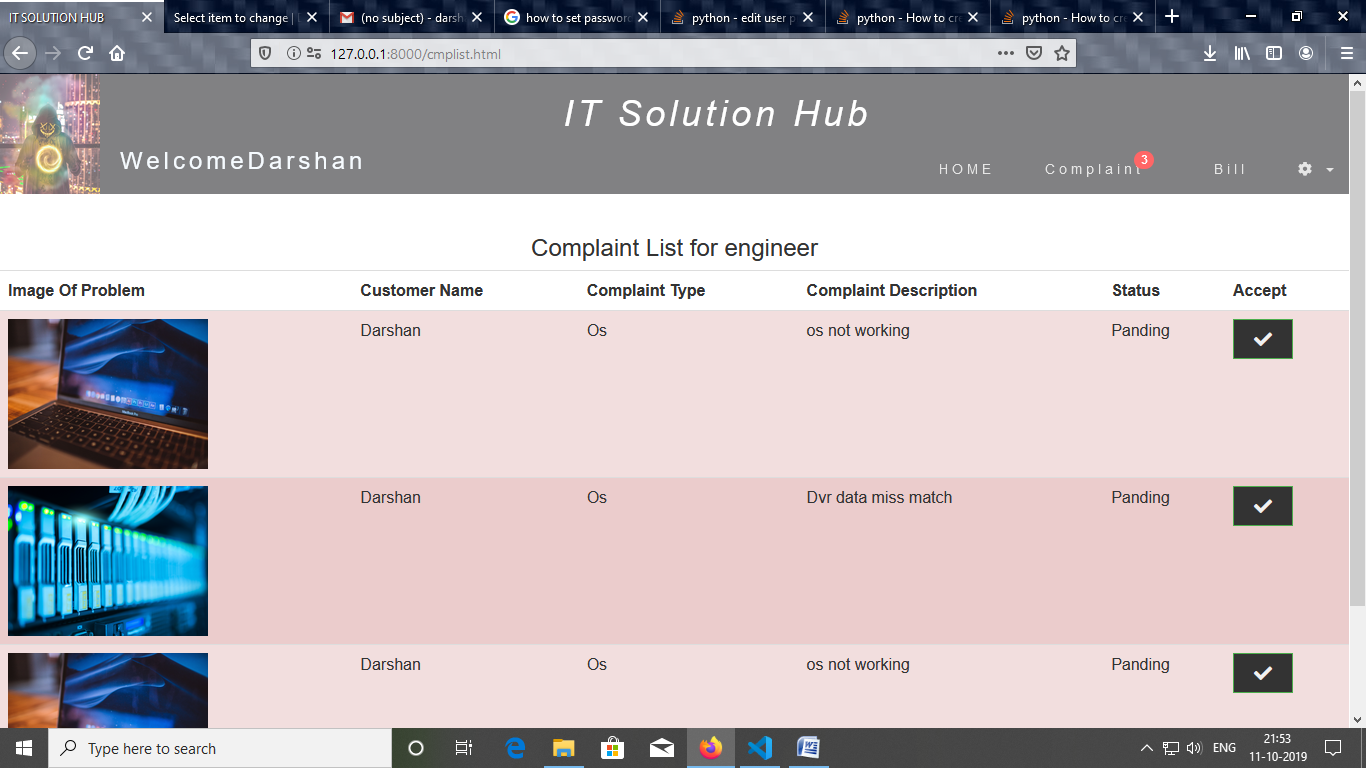


6.Product Rating Page

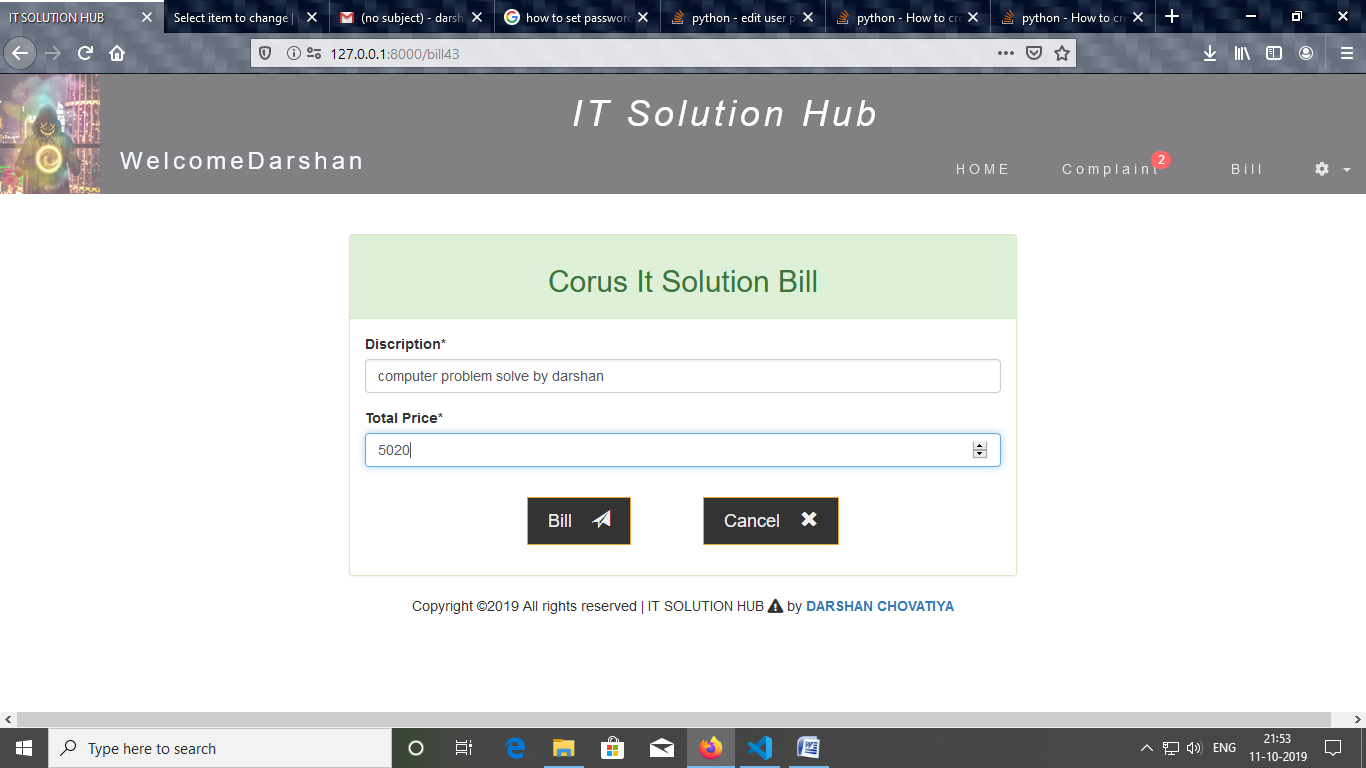


## 8.3. Engineer Login

1.Complaint Page

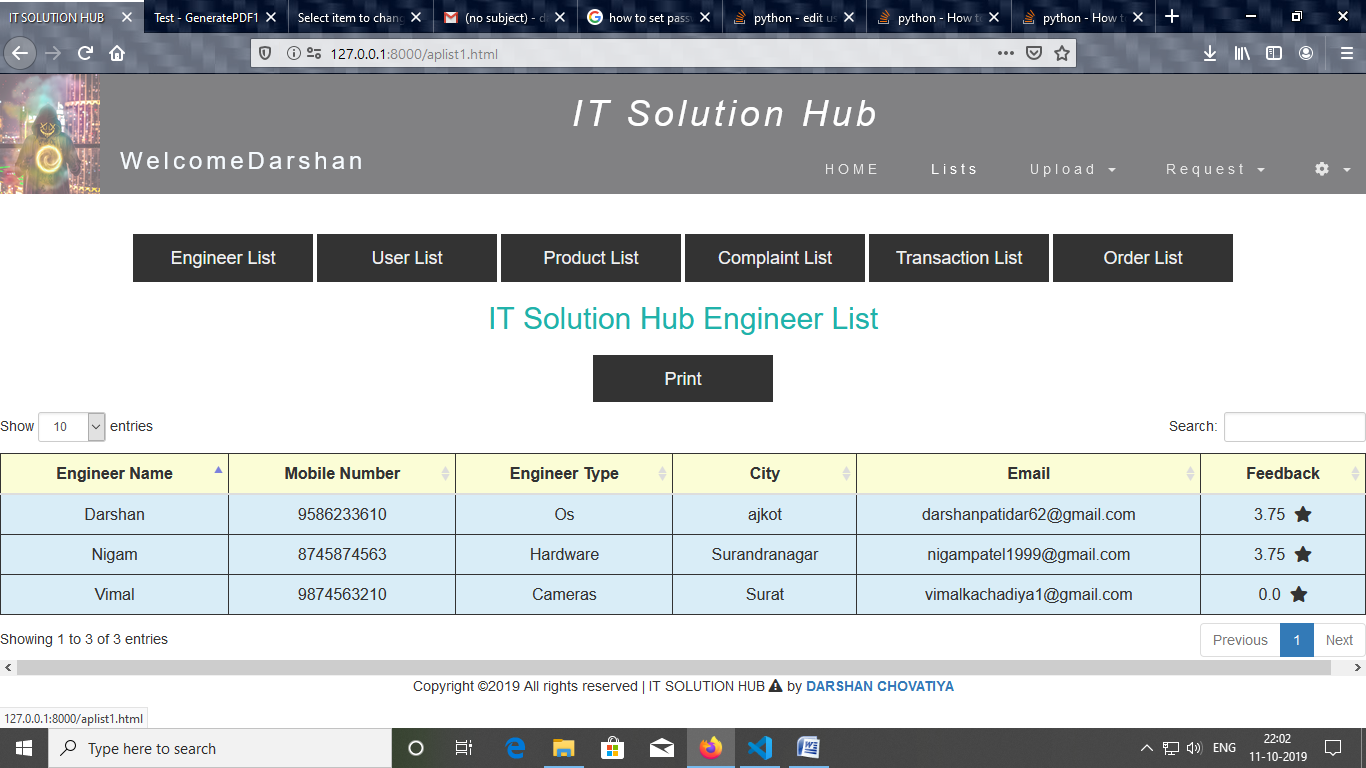


2.Complaint Bill Genarate Page

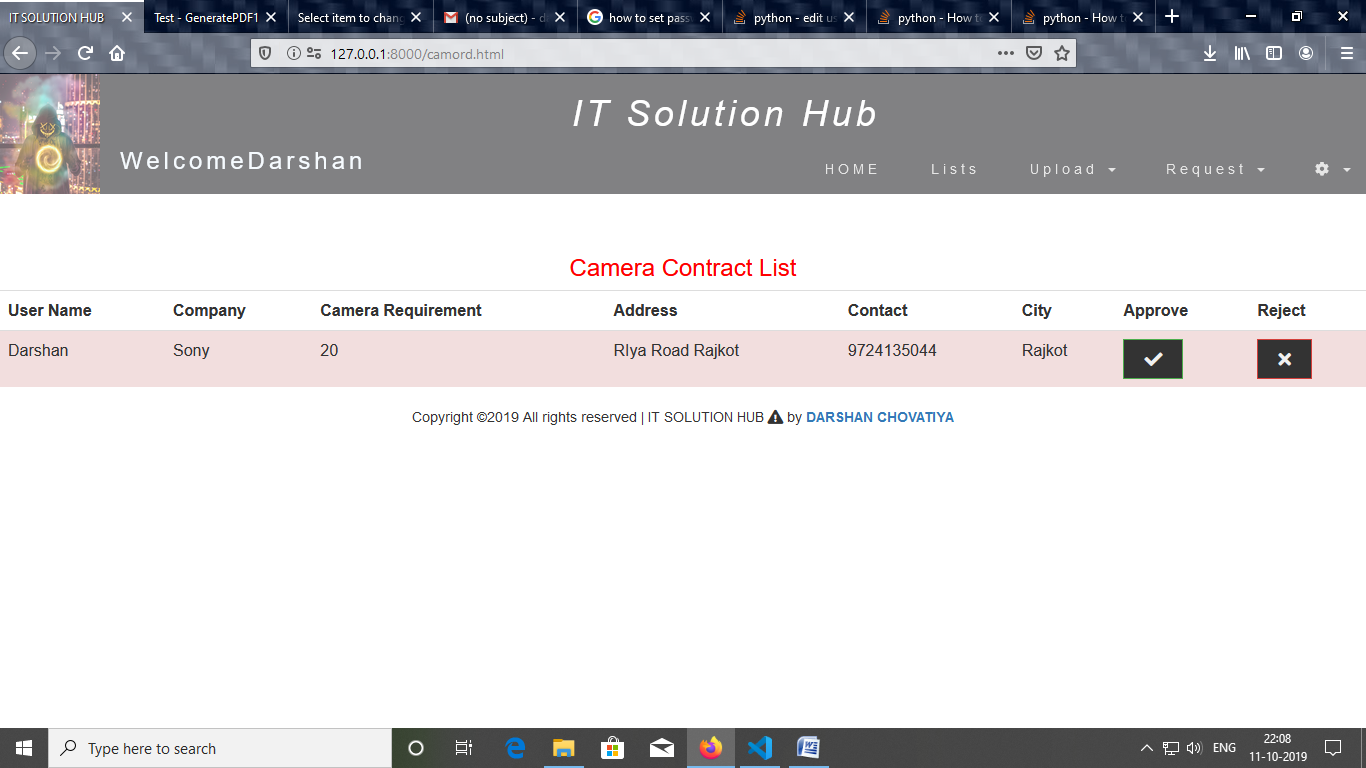


## 8.4. Admin Login

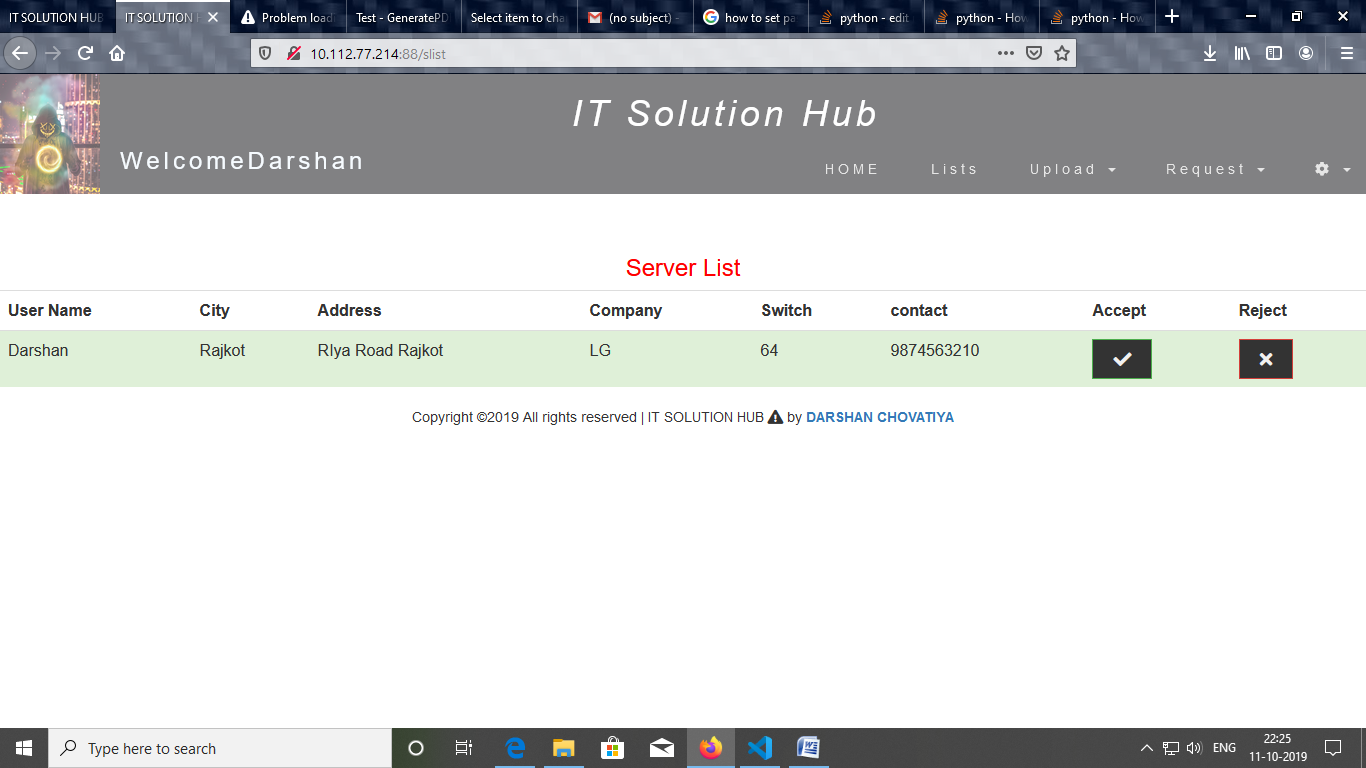
1. List all Tables



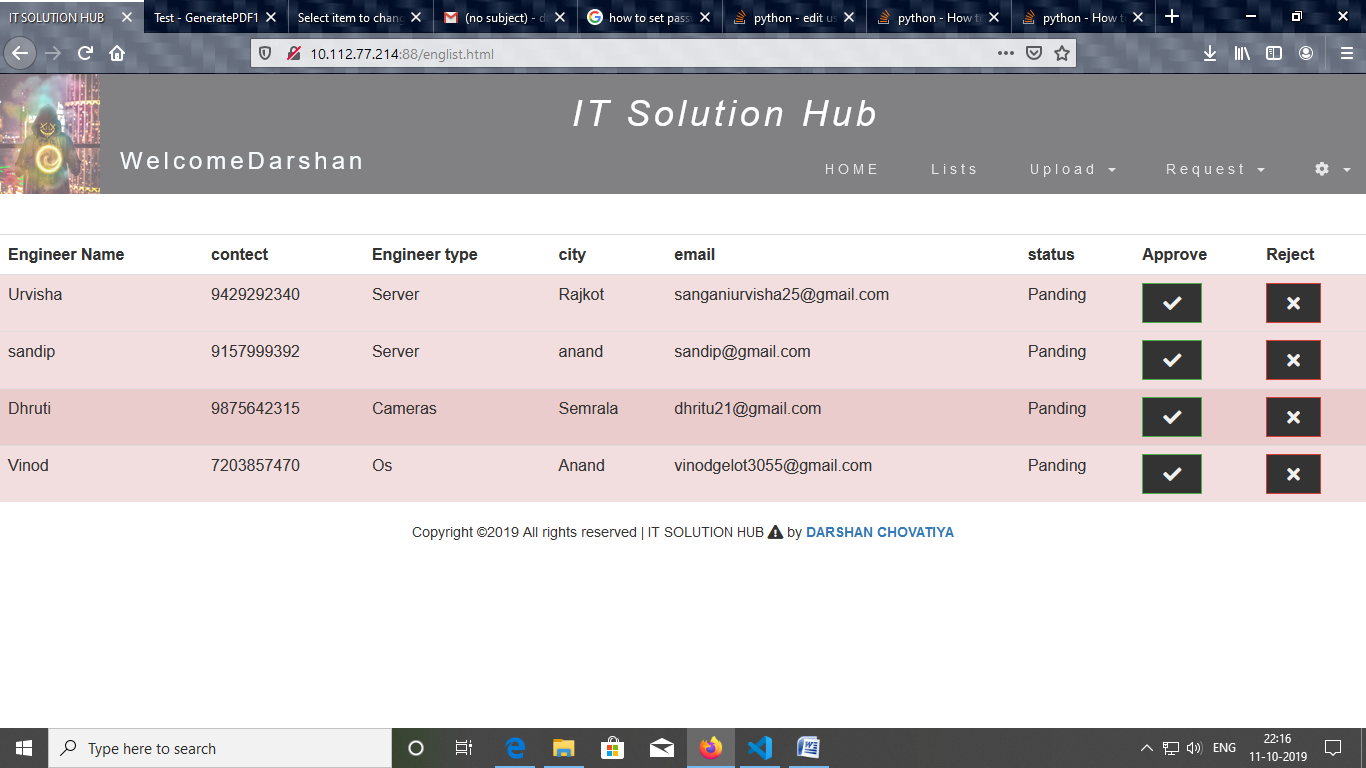
2.Camera Contract Page



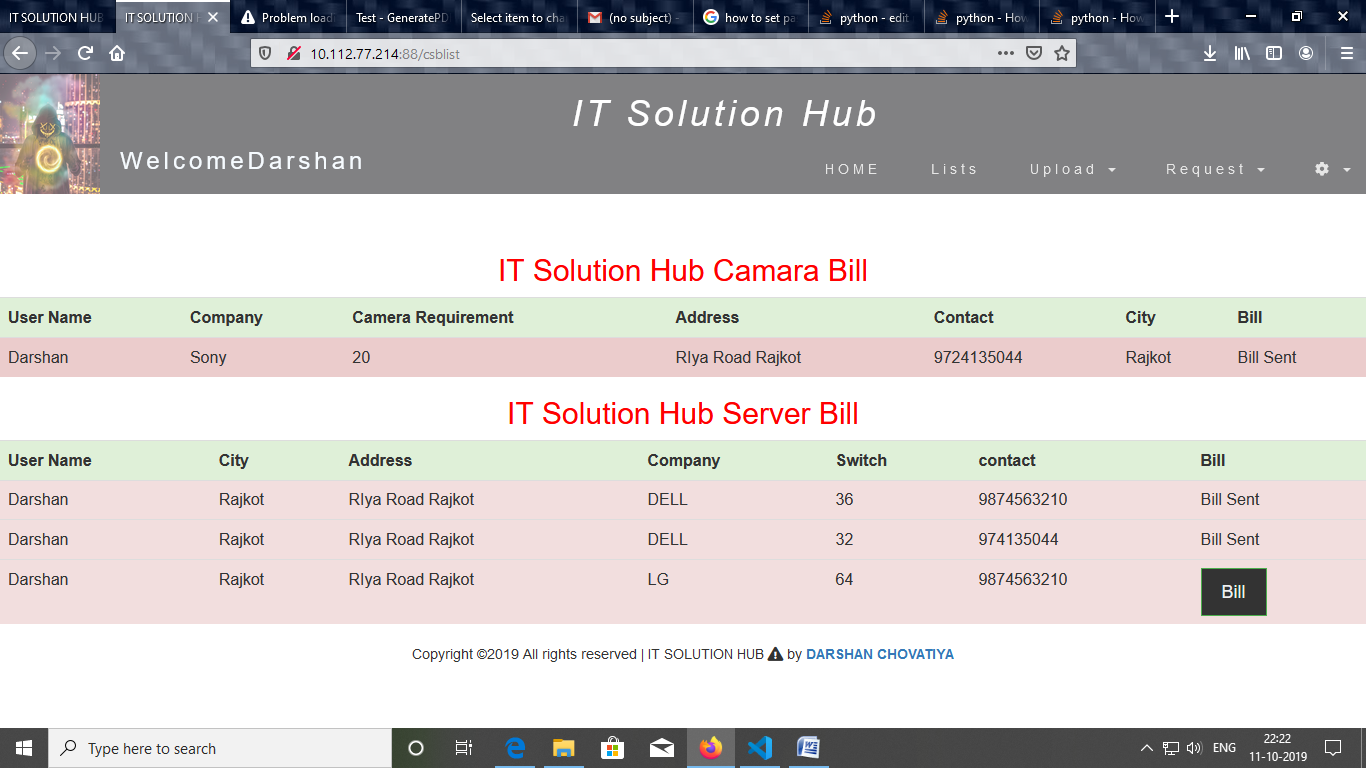
1. Server order Page



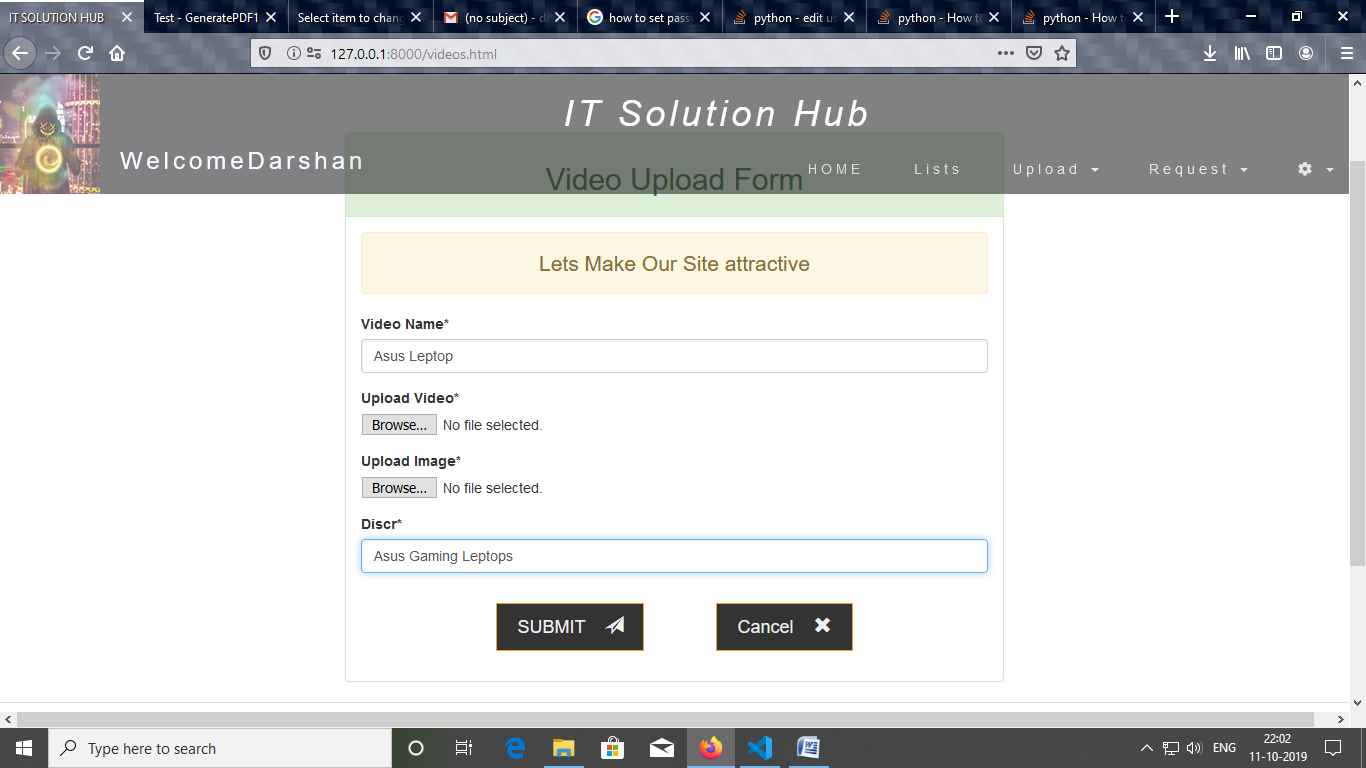
4. Engineer Login Requests Page



5. Camera and Server bill Page



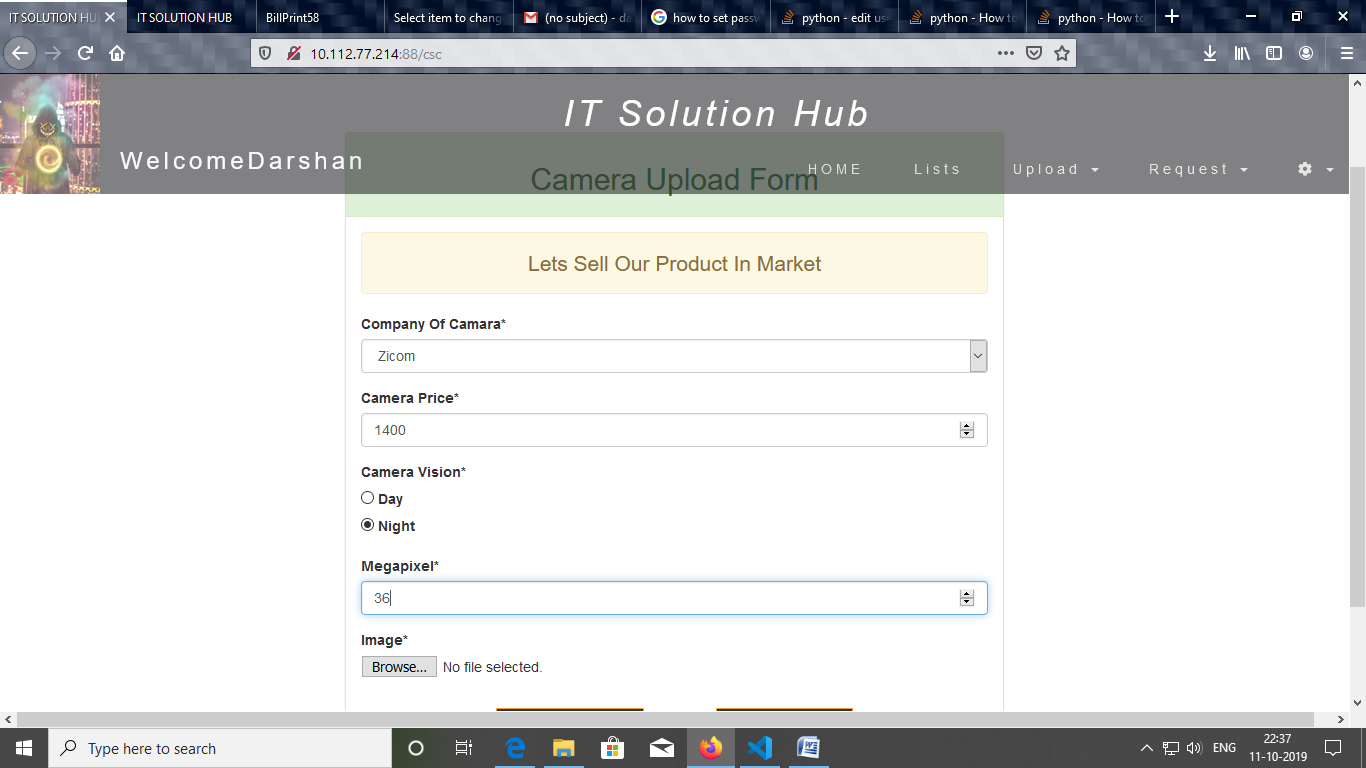
6. Video Upload form Page



7. Product Upload Page

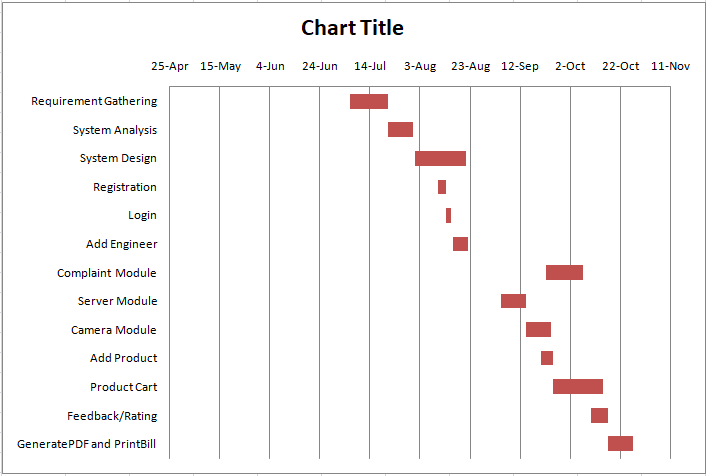


8. Camera Upload Page



# 9. Gantt Chart:

* **Process**
* Requirement Gathering(15 days)
* System Analysis(10 days)
* System Design(20 days)
* Registration(3 days)
* Login(2 days)
* Approve or Reject Engineer(6 days)
* Complaint Module(20 days)
* Server Module(10 days)
* Camera Module(10 days)
* Product Module(5 days)
* Shopping Cart(20 days)
* Feedback or Rating(7 days)
* Generate Report and Bill Print(10 days)



# 10. Conclusion:

The IT Solution Hub is web bases complaint and computer product selling system this system provides to a customer can send their problems related computer, server or camera the engineer can accept a request for complaint and he solves complain and generate the bill. Admin add all product, Camera and sell all list and accept the order of camera and server and also approve engineer, customer can add the product into the cart and buy the product and also give his/her feedback for a product and also engineer which solve his/her complaint.

11. References:

* <https://www.javatpoint.com/python-tutorial>
* <https://www.javatpoint.com/django-tutorial>
* <https://docs.djangoproject.com/en/2.2/>
* <http://www.learningaboutelectronics.com/Articles/How-to-create-a-video-uploader-with-Python-in-Django.php>
* <https://getbootstrap.com/>
* <https://stackoverflow.com/questions/21727317/how-to-check-confirm-password-field-in-form-without-reloading-page>
* <https://dev.to/djangotricks/how-to-create-pdf-documents-with-django-in-2019-5gb9>