**A**

**Project Report**

**On**

**DASHBOARD**

Developed by

**Chahak Agrawal – Department of IT, DD University**

**Mohil Kalavadia – Department of IT, DD University**

**Guided By**

**Internal Guide:**

**Prof. Ravindra.A.Vyas Department of Information Technology Faculty of Technology DD University**



**Department of Information Technology Faculty of Technology, Dharmsinh Desai University College Road, Nadiad-387001 April - 2020**

**CANDIDATE’S DECLARATION**

I declare that final semester report entitled “**Dashboard**” is my own work conducted underthe supervision of the internal guide **Prof. Ravindra.A.Vyas**.

.

I further declare that to the best of my knowledge the report for B.Tech. Final semester does not contain part of the work which has been submitted for the award of B.Tech. Degree either in this or any other university without proper citation.

Also I declare that following students also worked in this project:

Mohil Kalavadia – Department of IT, DD University

Chahak Agrawal– Department of IT, DD University

Mohil Kalavadia

Branch: H

Student ID: 16ITUOS068

Chahak Agrawal

Branch: H

Student ID: 16ITUOS065

**DHARMSINH DESAI UNIVERSITY**

**NADIAD-387001, GUJARAT**



**CERTIFICATE**

**This is to certify that the project entitled “Dashboard” is a bonafied report of the work carried out by**

**1) Mr. Mohil Kalavadia** **Student ID No: 16ITUOS068**

**2) Mr. Chahak Agrawal** **Student ID No: 16ITUOS065**

**of Department of Information Technology, semester VIII, under the guidance and supervision for the award of the degree of Bachelor of Technology at Dharmsinh Desai University, Nadiad (Gujarat). They were involved in Project training during academic year 2017-2018.**

Prof. Ravindra.A.Vyas,

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date:

Prof. (Dr.) V. K. Dabhi,

Head, Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad.

Date:

**TABLE OF CONTENTS**

ABSTRACT……………………………………………………………………………i

TABLES………………………………………………………………………………ii

1. List of Figures……………………………………………………………………....ii

2. List of Tables……………………………………………………………….……...iii

1. Introduction……………………………………………………………………….1
   1. Project Details…………………………………………………………….1
   2. Purpose……………………………………………………………………1
   3. Scope……………………………………………………………………...1
   4. Objective………………………………………………………………….2
   5. Technology and Literature Review……………………………………….2
2. Project Management……………………………………………………………..3
   1. Feasibility Study…………………………………………………………3

2.1.1 Technical feasibility……………………………………………3

2.1.2 Time schedule feasibility……………………………………….3

2.1.3 Operational feasibility………………………………………….3

2.1.4 Implementation feasibility……………………………………...3

* 1. Project Planning………………………………………………………….4
     1. Project Development Approach and Justification…………...4
     2. Project Plan…………………………………………………..5
     3. Milestones and Deliverables …………………………………5
     4. Roles and Responsibilities……………………………………6
  2. Project Scheduling…………………………………………………….....7

Project scheduling chart………………………………………………7

3.0 System Requirements Study……………………………………………………8

3.1 Study of Current System………………………………………………..8

3.2 Problems and Weaknesses of Current System………………………….8

3.3 User Characteristics…………………………………………………….8

3.4 Hardware and Software Requirements…………………………………8

3.5 Constraints……………………………………………………………...9

3.5.1 Hardware Limitations…………………………………………9

3.5.2 Interfaces to Other Applications………………………………9

3.5.3 Reliability Requirements...……………………………………9

3.5.4 Criticality of the Application………………………………….9

3.5.5 Security and Safety considerations……………………………9

3.6 Assumptions and Dependencies………………………………………9

4.0 System Analysis………………………………………………………………..10

4.1 Requirements of New System (SRS)…………………………………....10

4.3.1 User Requirements

* + 1. System Requirements
  1. Features Of New System………………………………………………...12

4.3 Navigation Chart…………………………………………………………12

* 1. Data Modeling
     1. ER Diagram……………………………………………….....13

1. System Design……………………………………………………………….14
   1. System Architecture Design…………………………………………….14
      1. Class Diagram……………………………………………….15
      2. Sequence Diagrams ………………………………………....16
      3. Component Diagram ………………………………………..18
      4. Deployment Diagram………………………………………..19
2. Implementation Planning………………………………………………………20
   1. Implementation Environment…………………………………………...20
   2. Program/Modules Specification………………………………………...20
3. Testing…………………………………………………………………………..21
   1. Testing Plan……………………………………………………………..21
   2. Testing Strategy………………………………………………………....21
   3. Testing Methods………………………………………………………...21
   4. Test Cases…………………………………………………………….....22
      1. Purpose
      2. Required Input
      3. Expected Result
4. User Manual……………………………………………………………………23
5. Limitation and Future Enhancement…………………………………………...30
6. Conclusion and Discussion…………………………………………………31
   1. Conclusions and Future Enhancement……………………………...31
   2. Discussion…………………………………………………………..31
      1. Self-Analysis of Project Viabilities…………………………31
      2. Problem Encountered and Possible Solutions………………31
      3. Summary of Project work…………………………………...31

11.0 References……………………………………………………………………..33

**ABSTRACT**

A data dashboard is an information management tool that visually tracks, analyzes and displays key performance indicators (KPI), metrics and key data points to monitor the health of a business, department or specific process. They are customizable to meet the specific needs of a department and company. Behind the scenes, a dashboard connects to your files, attachments, services and API’s, but on the surface displays all this data in the form of tables, line charts, bar charts and gauges. A data dashboard is the most efficient way to track multiple data sources because it provides a central location for businesses to monitor and analyze performance. Real-time monitoring reduces the hours of analyzing and long line of communication that previously challenged businesses.

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **NAME** | **PAGE** |
| Fig. 2.1 Iterative Waterfall Model | 4 |
| Fig. 4.1 Use Case Diagram | 12 |
| Fig 4.2 E-R Diagram | 13 |
| Fig. 5.1.1a MVC Design Model | 14 |
| Fig. 5.1.1b MVT Design Model | 14 |
| Fig. 5.1.3 Class Diagram (MVC Based) | 15 |
| Fig. 5.1.4 Sequence Diagram (MVC Based) | 16 |
| Fig 5.1.5 Sequence Diagram | 17 |
| Fig 5.1.6 Component Diagram | 18 |
| Fig. 8.1 Registration Page | 22 |
| Fig 8.2 Login Page | 23 |
| Fig 8.3 Dashboard Page | 24 |
| Fig 8.4 Dashboard Page | 24 |
| Fig 8.5 Dashboard Page | 25 |
| Fig 8.6 Dashboard Page | 25 |

**LIST OF TABLES**

|  |  |
| --- | --- |
| **NAME** | **PAGE** |
| Table 2.1 Milestones and Deliverables | 6 |
| Table 2.2 Roles and Responsibilities | 6 |
| Table 7.4 Test Cases | 21 |

**1.0** **INTRODUCTION**

* 1. **PROJECT DETAILS**

A dashboard is a consolidated display of many worksheets and related information in a single place. It is used to compare and monitor a variety of data simultaneously. The different data views are displayed all at once. Dashboards are shown as tabs at the bottom of the workbook and they usually get updated with the most recent data from the data source.

* 1. **MOTIVATION AND PURPOSE**

The whole point of the web-based dashboard is that it lets you visualize the Key Performance Indicators and other strategic data for your organization at a glance. It is the dashboard tool that presents management with the information for the practical end of the organization. Key Performance Indicators are quantifiable measurements, agreed to beforehand, that reflect the critical success factors of an organization. The web dashboard allows the user to view KPIs and other critical data without delving into the semantics of the source system that manages the detailed data

* + - Upload File
    - Create multiple dashboard for data analysis.
    - Make a new user account and give a secure login page.
  1. **SCOPE**

This website provides user to create theirs account and have profile of name email. Because of its conventional User Interface, it is necessity of login requirements or tribulation of accessing website or application. The system also can be further modified to have various functionalities under one platform.

* 1. **OBJECTIVES**

The Ultimate goal of the system is to give users a platform where he/she can represent data in form of different charts in order to analyze it. Here system provides different filters that can ease the data visualization.

* 1. **TECHNOLOGY AND LITERATURE REVIEW**
     + **Python:**

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scale. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

* + - **SDK:**

SDK (Software Development Kit) is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

* + - **Django:**

The Model-View-Template (MVT) is slightly different from MVC. In fact, the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL). The developer provides the Model, the view and the template then just maps it to a URL and Django

**2.0 PROJECT MANAGEMENT**

* 1. **FEASIBILITY STUDY**
     1. **Technical feasibility**

Technical analysis evaluates technical merits of the system at the same time collects additional information about performance, reliability, maintainability and productivity. The technical feasibility means that the project can be done with the current equipment, existing software tech- nology and the current knowledge. The present system is technically feasible as it has been developed on Python IDE - Atom for profes- sional developers and Django MVT framework for Web development. We are using Channels an easy-to-use Python library for having asynchronous messaging.

* + 1. **Time Schedule Feasibility**

The Project has simple working and the basic requirement can be satis- fied within allotted time period so the time development feasibility is satisfied.

* + 1. **Operational Feasibility**

There are mainly two functions: Creating dashboard and adding charts on that dashboard. The project is undertaken in the same environ using MVC-MVT framework of Django. Each operation has been working completely as an individual and so that the operational feasibility is satisfied.

* + 1. **Implementation Feasibility**

The system can be easily implemented as it has been developed in Atom Text Editor integrated with Django MVT framework. Implementation feasibility is concerned with specifying external resources and software that will successfully satisfy the requirements.

* 1. **PROJECT PLANNING**
     1. **Project Development approach and justification.**

For Project Development Iterative Waterfall Model is used. Iterative waterfall model:

The Iterative water fall model approach overcomes the problems associated with the waterfall model approach. If any difficulty or problem encounter in any phase may require going back to the previous phase and performing the required modifications and proceeds sequentially. This backtracking allows modifying any corrections or modifications required in the previous phase.

As illustrated in Fig 2.1, this model divides the cycle into the phases mentioned below:

* + - 1. Feasibility Study.
      2. Requirements analysis and specification.
      3. Design.
      4. Coding and Unit Testing.
      5. Integration and System Testing.
      6. Maintenance.

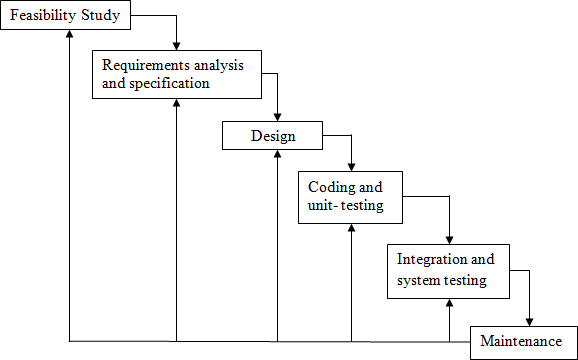


Fig 2.1 Iterative Waterfall Model

**Advantages of using Iterative Waterfall Model:**

* You are provided the chance to see the potential outcomes of every stage and make changes to areas of concern if necessary. This is one of the reasons that make the iterative model useful.
* Iterative development is more adjustable to changes as it considers each stage like a vital portion of the development cycle.
* The time spent on each successive interval may be lessened de- pending on how the last stage went and what knowledge was gained from past stages. The system therefore grows through adding new functionalities in the development part of all iterations.

**Disadvantages of using Iterative Waterfall Model:**

* When using the iterative model people working on the project can get stuck in a loop. Always finding problems than having to go back and design a fix, implement it, then test the system again and finding another problem can mean that the project can run over time and budget.
* Informal requests for improvement after each phase may lead to confusion and may also create scope creep, since user feedback following each phase may lead to increased customer demands. As users see the system develop, they may realize the potential of other system capabilities which would enhance their work, this can be an advantage as much as it can be a disadvantage.

**Justification:**

After feasibility study as the functional requirements were almost clear, but UI related requirements were not clear. Here we have decomposed the system into modules. That is why we decided to use iterative waterfall model which is most suitable model here i.e. if we find any difficulty in coding and testing a modification in design can be done easily.

* + 1. **Project Plan**

After feasibility study as the functional requirements were almost clear which were decided by our project lead. After analyzing and thor- oughly understanding the requirements of the application we planned the project.

The project had been planned in mainly two phase: Designing and Im- plementation. Designing phase consists of the road map of the system. It has been determined which frameworks and language gives optimal solution for the system and then in Implementation, it has been used to successfully accomplish the project.

* + 1. **Milestones and Deliverables**

Timely directions are always required to run a project successfully. Milestones tell the developers how far he has reached and also tell him what things are still left and how to fulfil them. Milestones may be the short report of achievement in project activity that are used by the

Project manager to check project progress but which are not delivered to the Clients. The deliverables are the project results that are provided to the customer. It is usually delivered at the end of some major project phases.

|  |  |  |
| --- | --- | --- |
| **MILESTONES** | **DELIVERABLES** | **PURPOSE** |
| Software Installation and Understanding of Technology. | Had complete knowledge of Django and its features. | To be familiar with Atom IDE. |
| System feasibility study, Requirement and Analysis. | Functional Specifications.  Non Functional Specifications. | It gives exact understanding of the User’s requirements. |
| System Design. | Dataset Design Class diagram  Sequence Diagram Use Case Diagram Form Layouts | It gives the logical Structure that describes the system. |
| Coding and Unit Testing and corrections if any. | Individually Tested and Functional Modules. Individual Modules for Sentiment Analysis | It gives the required Module. |
| Integration and System Testing. | The output obtained for the required functionality after implementing and doing various types of testing. | Integrated System is Ready. |

Table 2.1 Milestones and Deliverables

* + 1. **Roles and Responsibilities**

As only one members were involved in the whole team that individual had to perform all the tasks as the project proceeded through its different phases. This helped each one to develop skills in all the phases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Role | | | | |
| Analysis | Designing | Coding | Testing | Documentation |
| Mohil |  |  |  |  |  |
| Chahak |  |  |  |  |  |

* 1. **PROJECT SCHEDULING**

Scheduling the project tasks is an important project planning activity. It involves deciding which tasks should be taken up and when. In order to schedule the project activities; a software project manager needs to do the following:

* Identify all the tasks needed to complete the project.
* Break down large tasks into small activities.
* Determine dependencies amongst different activities.
* Establish most likely estimates for the time durations necessary to complete the activities.
* Allocate resources to activities.
* Plan the starting and ending dates for various activities.

**3.0 SYSTEM REQUIREMENTS STUDY**

* 1. **STUDY OF CURRENT SYSTEM**

The current website is good with creating charts in order to visualize data. In this system we can create a dashboard and upload any type of file of which desired charts are prepared for analysis.

* 1. **PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM**

Current website dashboard is good with managing data from source and representing them in form of chart. But it is not having enough filter to be applied on charts for better analysis.

* 1. **USER CHARACTERISTICS**

The targeted users can be any individual who wants to analyze data in dashboard. Also, user can create charts of their choice. One can also create multiple dashboard at a time.

* 1. **HARDWARE AND SOFTWARE REQUIREMENTS**
     1. **Software requirements**
        + **Server Side:**

User interface: Django Web Application Documentation Tool: Word

IDE: platformio

* + - * **Client Side:**

The user’s browser should support HTML5, Bootstrap and Ja- vaScript for a satisfactory user experience.

* + 1. **Hardware requirements**

The device should be enabled with the Internet

* 1. **CONSTRAINTS**
     1. **Hardware limitations**

There are no hardware limitations for this system because once the complete system is developed care would be taken while deploying system so necessary pre-requisites are met.

* + 1. **Interface to other applications**

There are no other systems that use this application as an interface.

* + 1. **Reliability Requirements**

The application does demand much reliability and it is fully assured that the particular information about the user should be secured and flow is maintained and accessed according to the rights.

* + 1. **Criticality of the Application**

The application deals with the account with authentication and currently run on the localhost. So No need to worry about criticality.

* + 1. **Security and Safety considerations**

The system provides the tight security for information of user and is secured at the developer end.

* 1. **ASSUMPTIONS AND DEPENDENCIES**

Assumptions are described as follows:

* User has sufficient privileges to access internet.
* Server is running smoothly.

**4.0 SYSTEM ANALYSIS**

* 1. **REQUIREMENTS OF NEW SYSTEM**
     1. **User Requirements**

**Description:** System renders the front page of the website on the browser.

* + 1. **System Requirements**
       1. **Login**

**Input1:** login credentials

**Output:** Login successful/failed

**Description:** After the successful login the user is redirected to the home page of the website.

* + - 1. **Registration**

**Input:** User details

**Output:** successfully registered

**Description:** After the user has successfully given the correct details and valid password string it is successfully registered and account has been created of the user.

* + 1. **Create Dashboard**

**Input:** Dashboard Detai**l (**dash Name, description**)**

**Output:** Created successfully

**Description:** the user can create dashboard. After entering name and description of the group. User can add graph inside that dashboard of any data.

* + 1. **Add Charts**

**Input:** File, dashboard, Type, x-axis, y-axis

**Output:**  added successfully

**Description:** the user can add graph in any Dashboard he/she want to. User can add chart of any type from any data file.

* + 1. **Upload File**

**Input:** filename

**Output:** File upload

**Description:** User can upload any file he/she want in order to analyze its data by plotting charts and adding it in dashboards.

* 1. **FEATURES OF NEW SYSTEM**

The new system consists the features of user interactive functionality along with existing features and more functionality combined.

* 1. **SYSTEM ACTIVITY (USE CASE OR SCENARIO DIA- GRAM)**



Fig. 4.1 System Activity (Use Case diagram)

* 1. **Entity-Relationship Diagram**

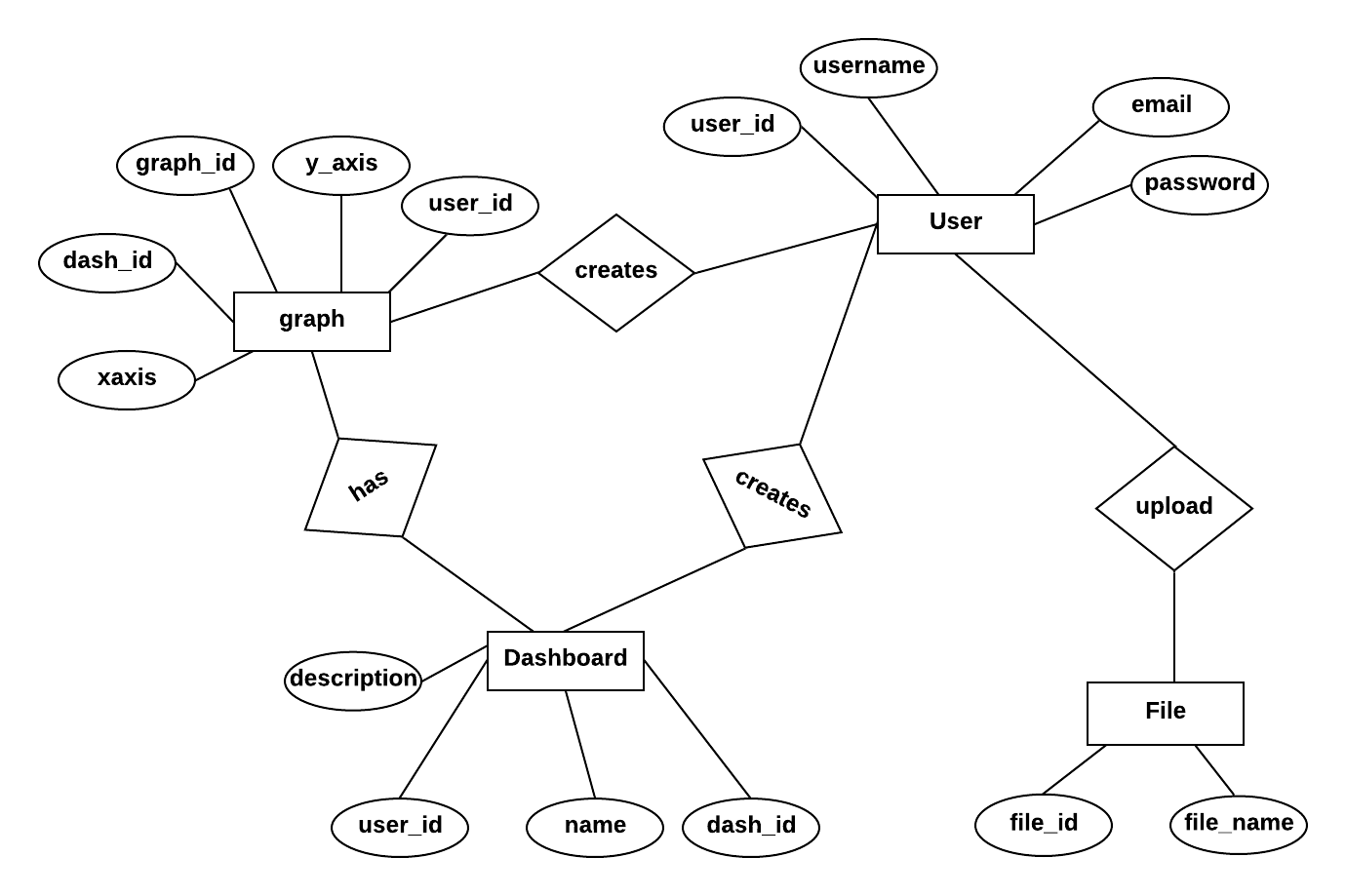


Fig. 4.2 E-R Diagram

**5.0 SYSTEM DESIGN**

* 1. **SYSTEM ARCHITECTURE DESIGN**
     1. **MVC-MVT Design Model**

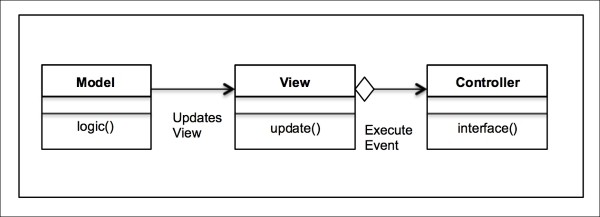


Fig. 5.1.1a MVC Design Model

* + - * Model: Python file for Dashboard
      * View: Django framework
      * Controller: URL

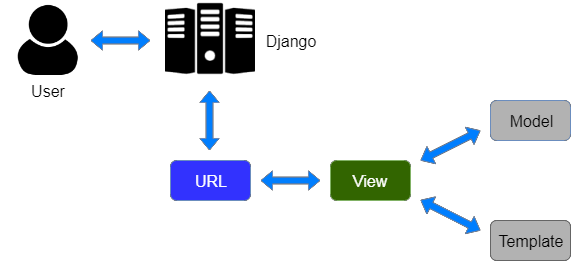
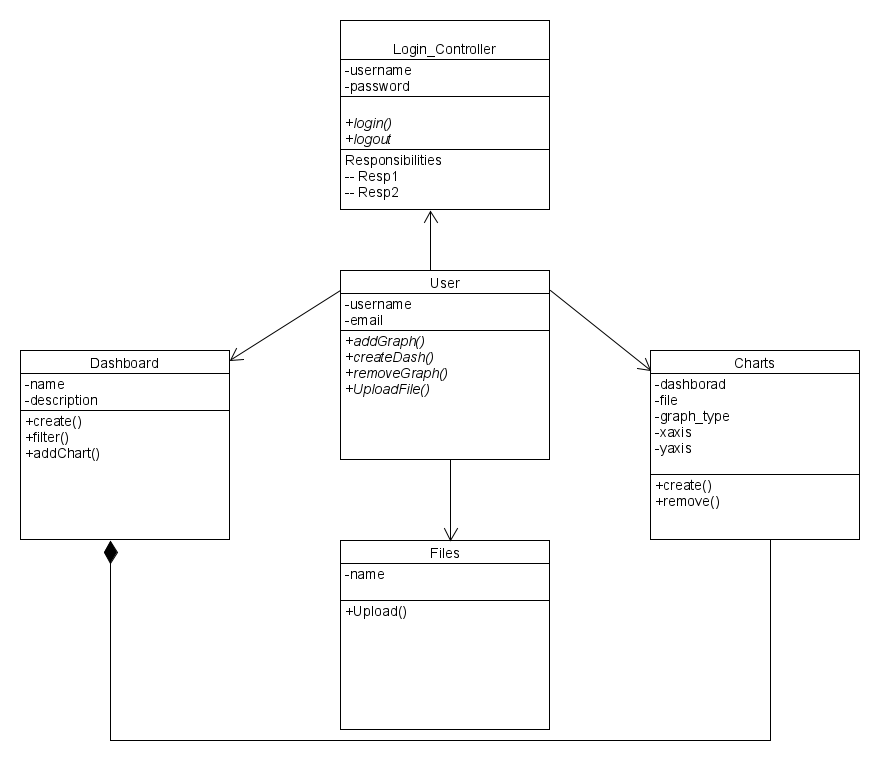


Fig. 5.1.1b MVT Design Model

* + 1. **Class Diagram (MVC Based)**

Fig. 5.1.3 Class Diagram (MVC Based)



* + 1. **Sequence Diagram (MVC Based)**

1. **Sequence Diagram for Create Dashboard**

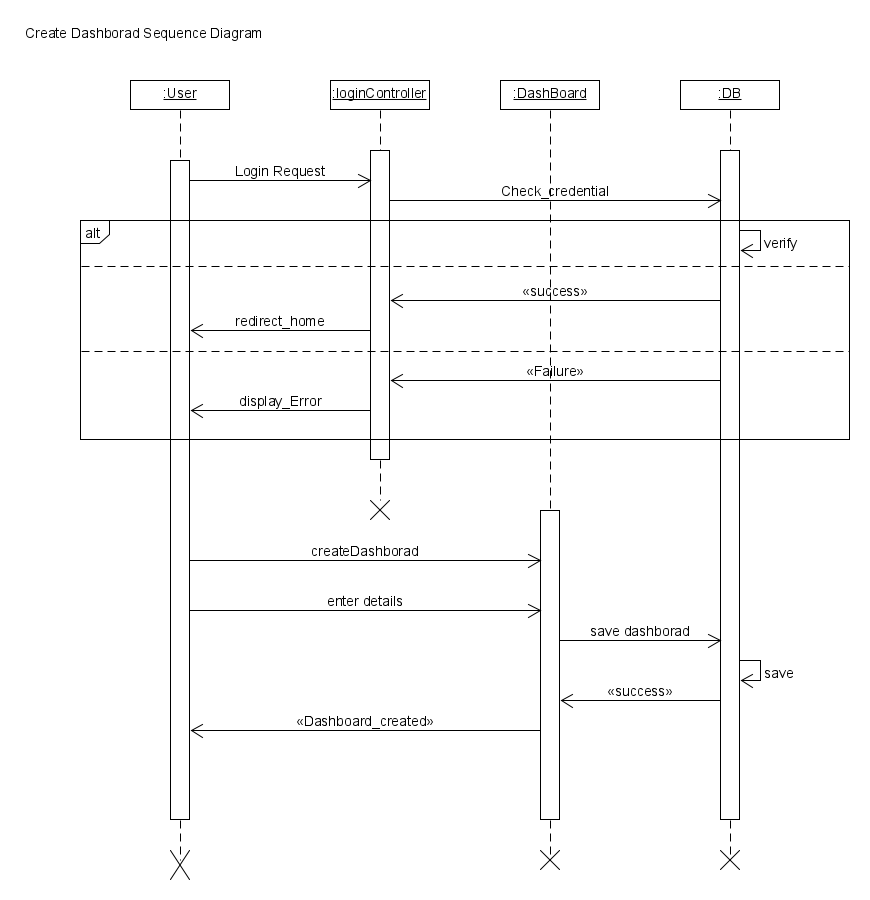


Fig. 5.1.4 Sequence Diagram

**2) Sequence Diagram for Add Graph**

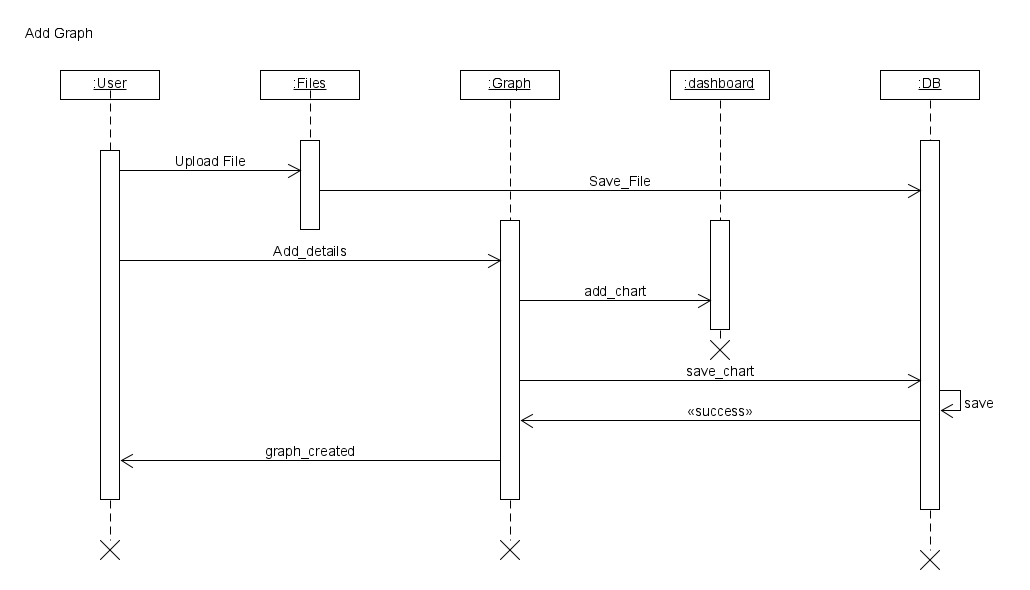
****

Fig. 5.1.5 Sequence Diagram

* + 1. **Component Diagram**

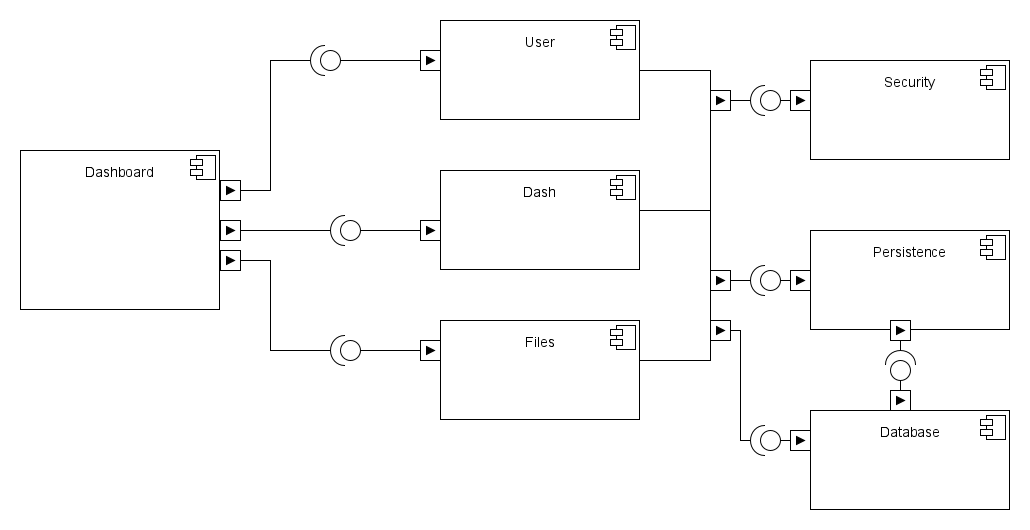
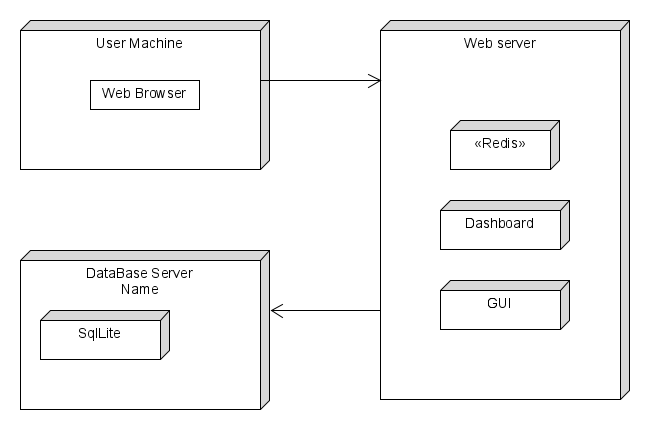


Fig. 5.1.6 Sequence Diagram

**5.1.5 Deployment Diagram**



**6.0 IMPLEMENTATION PLANNING**

* 1. **IMPLEMENTATION ENVIRONMENT**

The application is a multiuser system with GUI. For the implementation of the project, the environment required is Windows 10. The language used are Python. Framework used are Django 2.0.

* 1. **PROGRAMS / MODULES SPECIFICATION**

Atom Text Editor is used for implement and debugging the program of the website. Platformio terminal is used for running and install modules required for application.

* + 1. **Django-Channels**

Channels changes Django to weave asynchronous code underneath and through Django’s synchronous core, allowing Django projects to handle not only HTTP, but protocols that require long-running connections too - Web Sockets, MQTT, chatbots, amateur radio, and more.Channels is a project that takes Django and extends its abilities beyond HTTP - to handle Web Sockets, chat protocols, IoT protocols, and more. It’s built on a Python specification called ASGI. It does this by taking the core of Django and layering a fully asynchronous layer underneath, running Django itself in a synchronous mode but handling connections and sockets asynchronously, and giving you the choice to write in either style.

Django channels’ goal is to extend the Django framework, adding to it a new layer to handle the use of Web Sockets and background tasks. It is used to implement chat functionality in our website.

* + 1. **Plotly**

[Plotly.js](https://plot.ly/javascript/) is a library ideally suited for JavaScript applications which make use of graphs and charts. There are a few reasons to consider using it for your next data visualization project. Plotly.js uses both D3.js (SVG) and WebGL for graphics rendering. Plotly.js is an “all-in-one bundle” with d3.js and stack.gl modules. It works with JSON schema. Plotly.js supports basic, statistical, scientific, financial and map charts.

**7.0 TESTING**

* 1. **TESTING PLAN**

The testing technique that is going to be used in the project is black box testing. In black box testing the expected inputs to the system are applied and only the outputs are checked.

* 1. **TESTING STRATEGY**

The development process repeats this testing sub-process a number of times for the following phases.

* + 1. Unit Testing.
    2. Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

* 1. **TESTING METHODS**

Black box and White box testing:

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behaviour. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing. For this purpose, tools or compilers that can insert test probes into the programs can be used.

* 1. **TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Description** | **Prerequisite** | **Expected Output** | **Result Output** | **Pass/Fail** |
| 1.1 | Server  Listening | The Port Number should be vacant | Successful deployment of server and listening for  request | Successful deployment and request accepted | Pass |
| 1.2 | Login | The user enter the credential to log into website | According to the credential appropriate login success or failure message should be displayed. | User logs in successfully. | Pass |
| 1.3 | Dashboard Creation | User has entered valid dashboard name and description and page is submitted  properly | Dashboard should be created with proper description | With valid Dashboard name ,the Dashboard is created along with its description | Pass |
| 1.4 | Graphs should be added into Dashboard | User has entered valid Dashboard name and charts are and page is submitted. | charts should be added into dash | charts added into dash | Pass |

Table 7.4 Test Cases

**8.0 USER MANUAL**

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system.

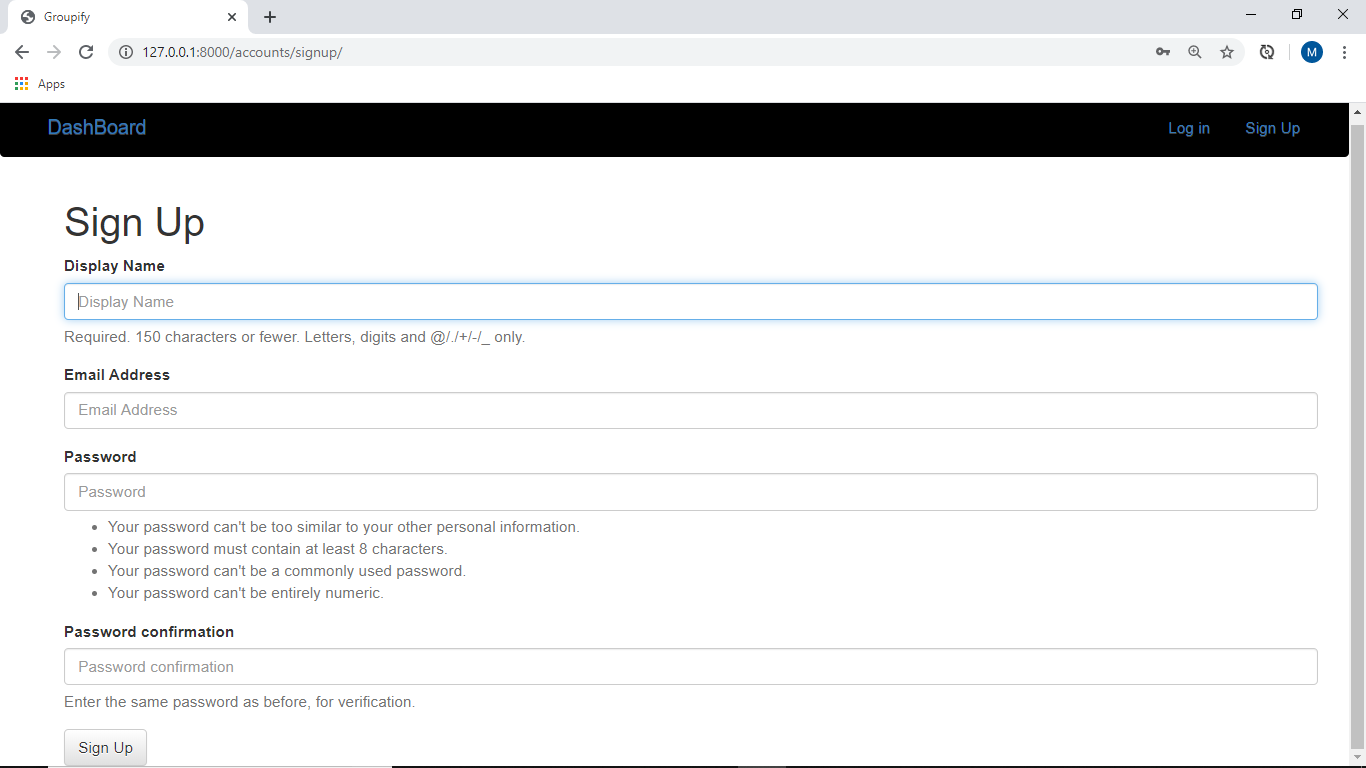


Fig. 8.1 Registration

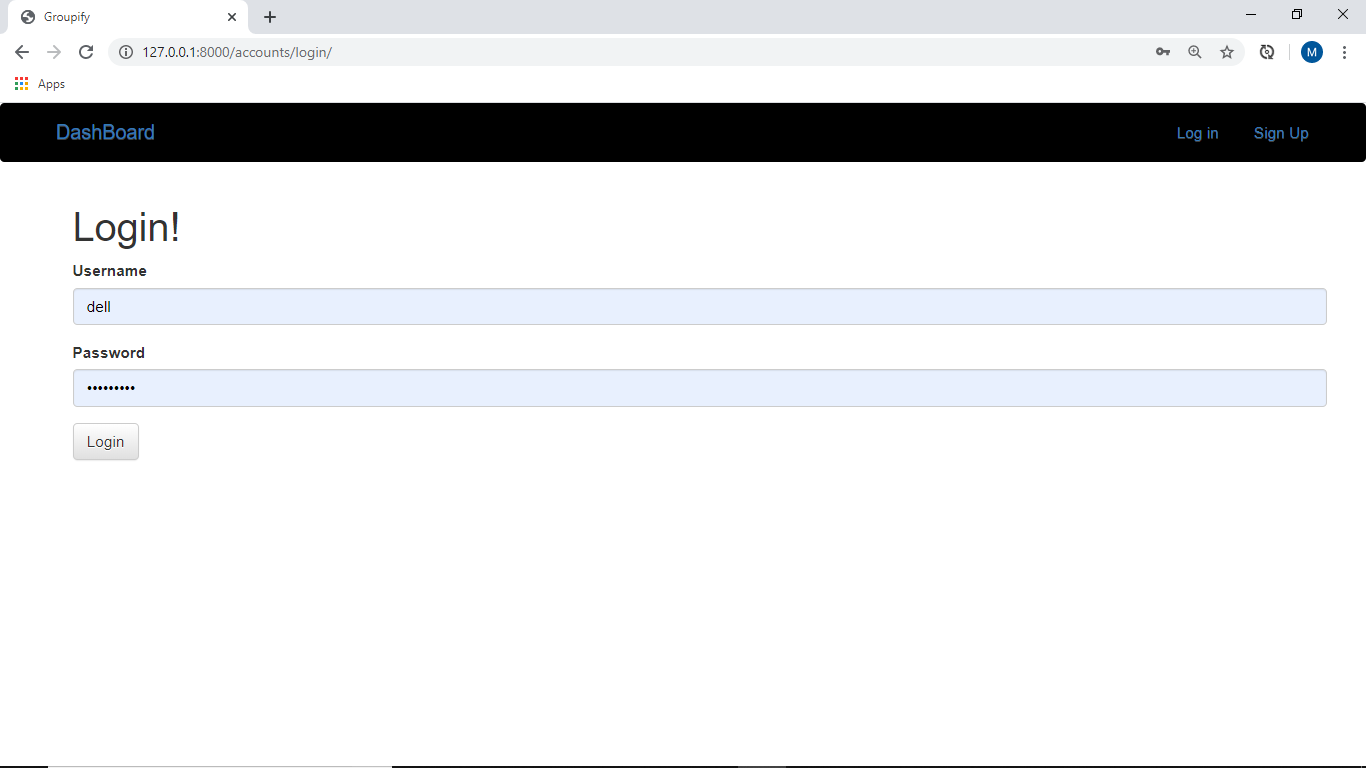


Fig. 8.2 Login Page

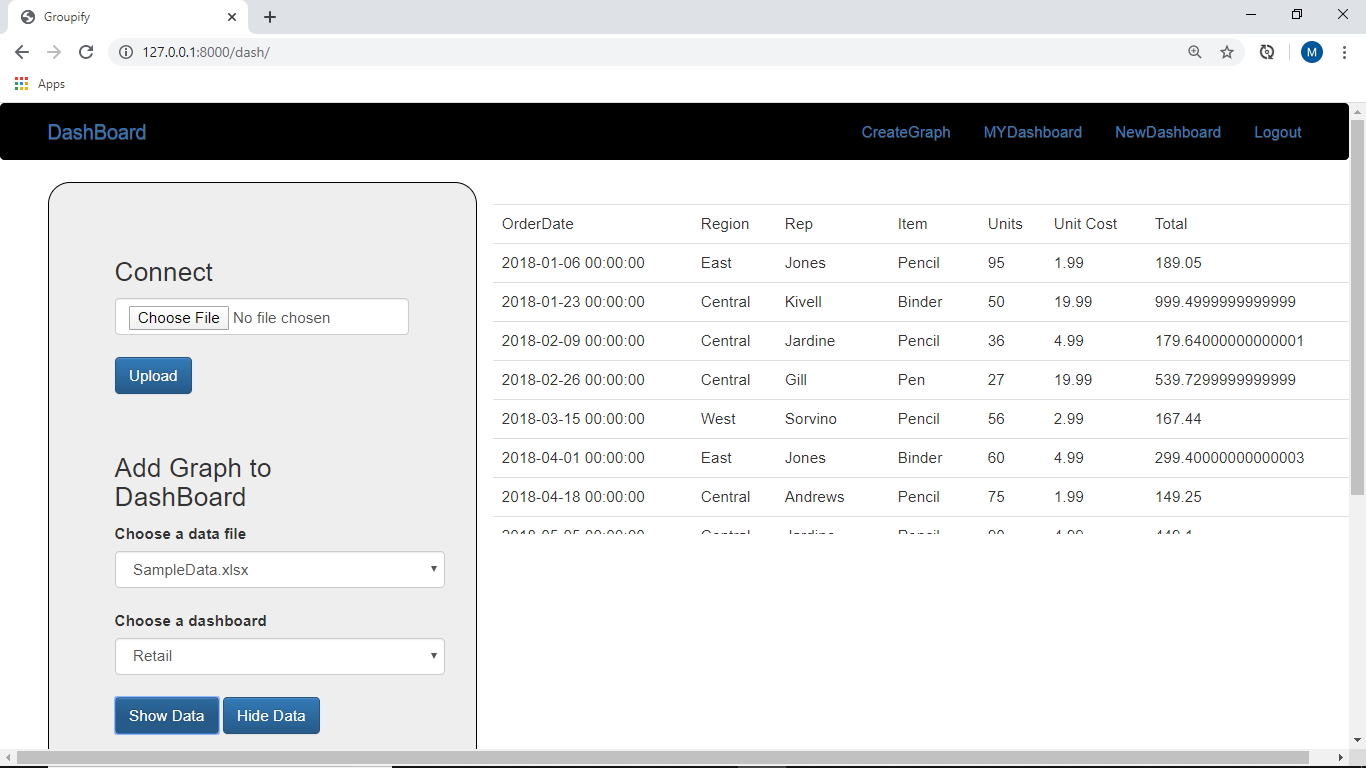


Fig. 8.3 Dash Page

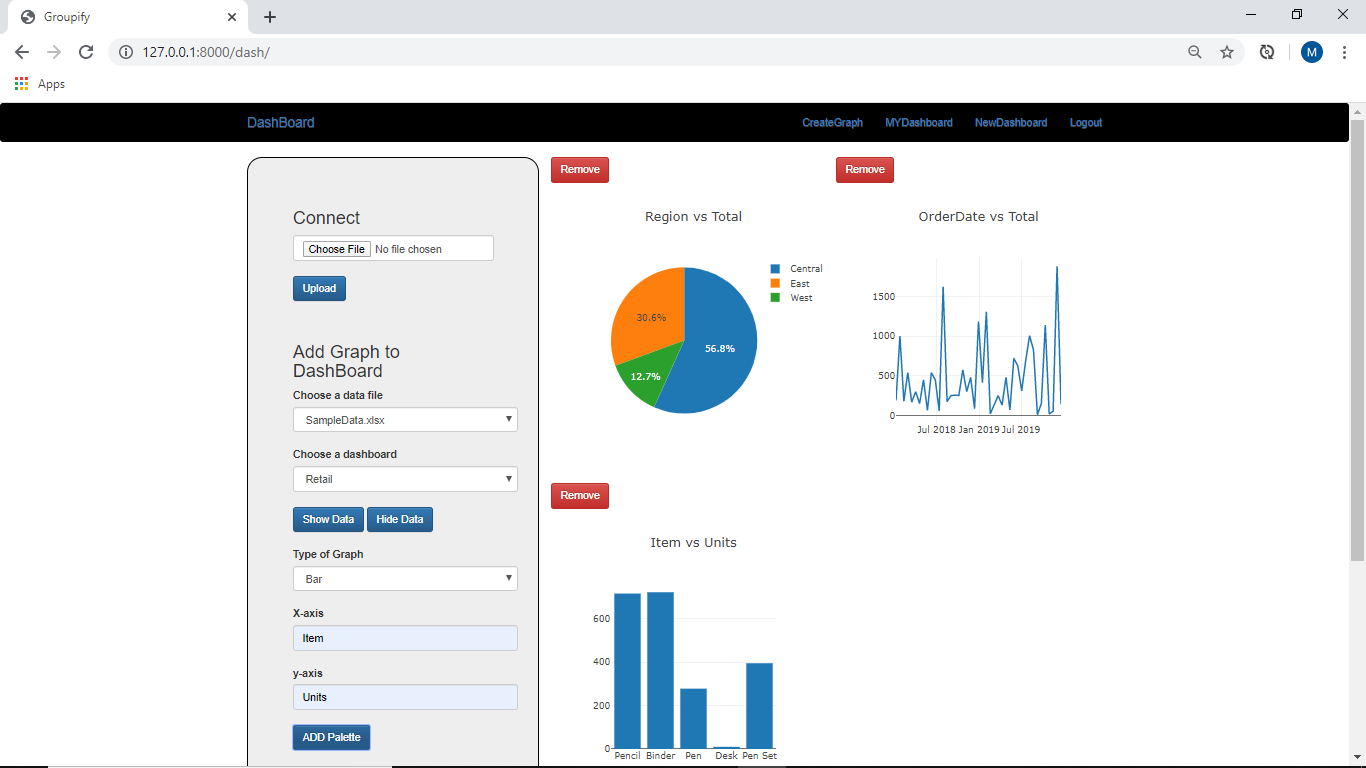


Fig. 8.4 Dash Page

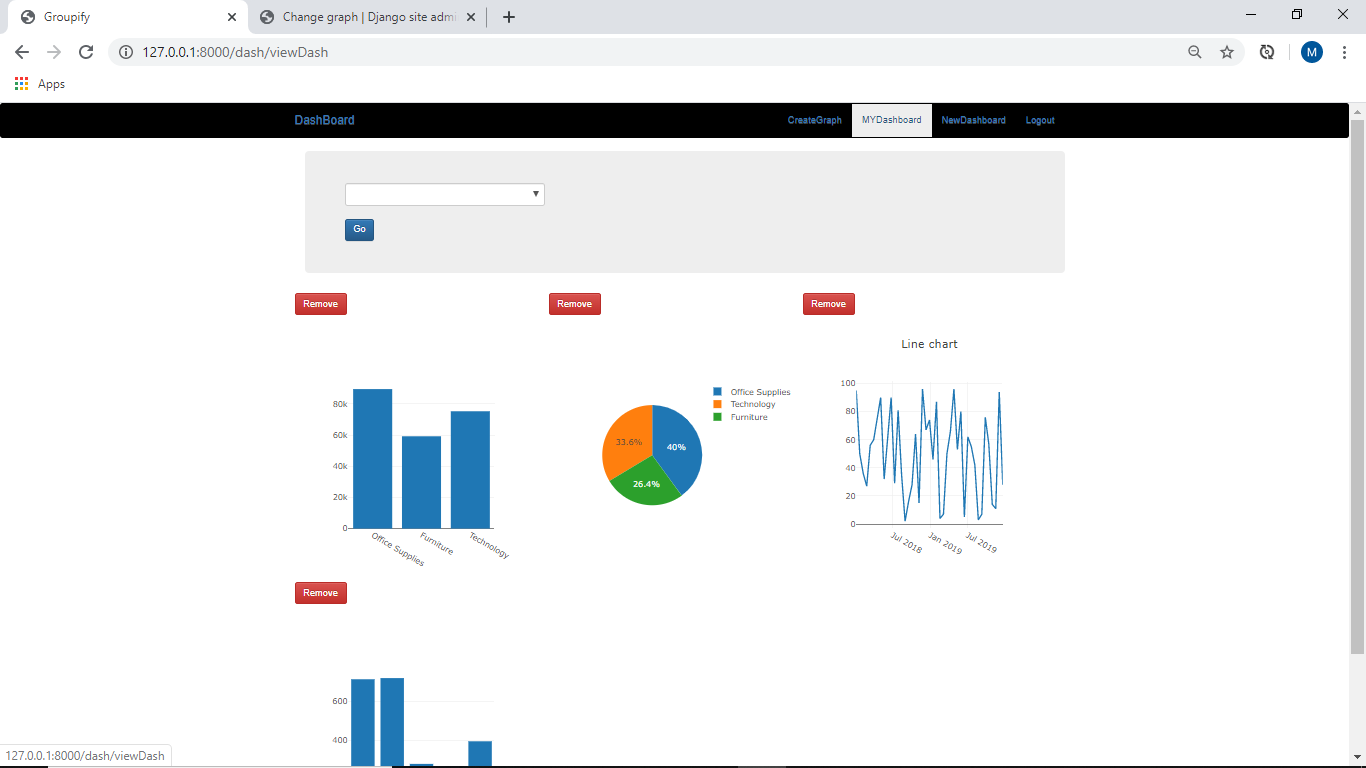


Fig. 8.5 Dash page

**9.0 LIMITATION AND FUTURE ENHANCEMENT**

* 1. **LIMITATIONS**

This project lacks some of the functionality that must be included to create a dashboard. This project is not able update automatically when there is change in data of the data source.

The system is not able to give more complete functionalities like present dashboard websites. The system is not have enough filters for chart manipulation.

* 1. **FUTURE ENHANCEMENT**

The system can further be improved by giving more functionalities to it. The functionalities like adding more filters and more type of graph. Further we could give more professional user interface to the system. The system can be expanded by using APIs for dynamically using the functionalities of present Dashboard.

**10.0 CONCLUSION AND DISCUSSION**

* 1. **CONCLUSION**

While developing a system conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources –that would generate a proper system for Dashboard.

While making this project, an eye has been kept on making it user friendly. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in case of any system development process where there are a number of short comings, there have been some shortcomings in the development of this system also. There are some the areas improvement which couldn’t be implemented due to time constraints. One such feature was giving connection to other database.

* 1. **DISCUSSION**
     1. **Self-Analysis of Project Viabilities**

According to us, this project is absolutely a good start for gaining hands-on experience on Django. It is useful if it is managed according to the goal for which it is made.

* + 1. **Problems Encountered and Possible Solutions**

Following are problems encountered during this project.

* Web Sockets
* Creating graph
  + 1. **Summary of Project Work**

It has been successfully completed the project. The prior knowledge of software engineering has helped immensely in overcoming the various roadblocks. We have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

## **11.0 REFERENCES**

1. https://django-private-chat.readthedocs.io/en/latest/
2. https://docs.djangoproject.com/en/2.2/
3. https://getbootstrap.com
4. https://plot.ly/javascript/