EVENT TRACKER – A DJANGO BASED WEB APPLICATION

A Online Summer Internship project report submitted to the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD** in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Saketh Reddy R(18071A0547)

Tammisetti Satya Sairam (18071A0555)

Vancha Srii Sarvotam Reddi (18071A0558)

Under the Guidance of

Mrs A. Madhavi

(Assistant Professor , VNR VJIET)

TAMASOMA JYOTHIRGAMAYA

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AN AUTONOMOUS INSTITUTE, NAAC ACCREDITED WITH 'A++' GRADE, NBA ACCREDITED, APPROVED BY AICTE, NEW DELHI, AFFILIATED TO JNTUH)

August,2020

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND

TECHNOLOGY

(AN AUTONOMOUS INSTITUTE, NAAC ACCREDITED WITH 'A++' GRADE, NBA

ACCREDITED, APPROVED BY AICTE, NEW DELHI, AFFILIATED TO JNTUH)

CERTIFICATE

This is to certify that R. Saketh Reddy (18071A0547),T. Satya Sairam (18071A0555) ,Vancha Srii Sarvotam Reddi (18071A0558) have successfully completed their Online Summer Internship project work at CSE Department of VNR VJIET, Hyderabad entitled **"Event Tracker – A Django Based Web Application"** in partial fulfillment of the requirements for the award of B.Tech degree during the academic year 2020-2021.

This work is carried out under my supervision and has not been submitted to any other University/Institute for award of any degree/diploma.

Mrs A. Madhavi   **Dr. B.V.Kiranmayee**

Project Guide   Associate Professor and Head

Assistant Professor   CSE Department

CSE Department   VNRVJIET.

VNRVJIET.

DECLARATION

We hereby declare that the mini project entitled "**Event Tracker – Django Based Web Application**" submitted in partial fulfilment of the requirements for award of the degree of Bachelor of Technology in Computer Science and Engineering at VNR Vignana Jyothi Institute of Engineering and Technology, affiliated to Jawaharlal Nehru Technological University, Hyderabad, is a bonafide report of the work carried out by us under the guidance and supervision of Mrs A .Madhavi (Assistant Professor), Department of CSE, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any University/Institute for award of any degree or diploma.

R. Saketh Reddy   **T. Satya Sairam**   **Vancha Srii Sarvotam Reddi**

(18071A0547)   (18071A0555)   (18071A0558)

III.BTECH CSE   III.BTECH CSE   III.BTECH CSE

VNRVJIET   VNR VJIET   VNR VJIET

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without it would ever never have come into existence. To them we lay the words of gratitude imprinting within us.

We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which lead us to complete this mini project. The support, encouragement given by him and his motivation lead us to complete this mini project. We are very much thankful to our H.O.D., **Mrs. B.V. Kiranmayee** madam for extending her cooperation in doing this project.

We extend our heartfelt thanks to our guide, **Mrs A. Madhavi** ma'am, for her enthusiastic guidance throughout the course of our project, also who extended her valuable support in helping us complete the project in a correct way.

Last but not the least, our appreciable obligation also goes to all staff members of Computer Science & Engineering Department and to our fellow classmates who directly or indirectly helped us.

<div align="right">

R. SAKETH REDDY(18071A0547)

T. SATYA SAIRAM(18071A0555)

VANCHA SRII SARVOTAM REDDI(18071A0558)

</div>

ABSTRACT

"In these days almost every college is conducting technical / non-technical events where we can gain some knowledge by participating. College, which is conducting events, this project can be organizer for the various events offered by them. This provides college management to schedule events online and assign organizers for an event. Event Organizers can upload information related to the event in the form of text, audio, video files. The students and participants can view these files online and download.

Event organizers create schedules for events using web app interface. The students register with the site and some of them register as coordinators. Only event organizers and volunteers can upload event content to the site. Event Tracker also provide facility to provide all information about all events which are being conducted. So, every student should get proper knowledge about events and he can participate if he wants.

Here comes the salient feature of the Event Tracker which is enabled by Django, it can automatically generate certificates and distribute them to the participants via Email and also provides facility for the students to download it from their dedicated account . This enables students to view their certificate and download them when ever require at any time in the career.

One of the main agenda of the Event Tracker is to centralize all the events happening in campus. An authority is given access to create organizer accounts so that any organizer should approach him to register his personalized account which he can use to track the student registrations , mark attendance and view event statistics."

Keywords : Event Tracker , Django , Web app interface.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Introduction to Web – Application:

A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet.

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.

The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.

Here's what a typical web application flow looks like:

1. **User** triggers a request to the **web server** over the **Internet**, either through a web browser or the application's user interface.
2. **Web server** forwards this request to the appropriate **web application server.**
3. **Web application server** performs the requested task – such as querying the **database** or processing the data – then generates the results of the requested data.
4. **Web application server** sends results to the **web server** with the requested information or processed data.
5. **Web server** responds back to the client with the requested information that then appears on the user's display.

1.2 Introduction to Django:

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.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.[7] Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views",[5] the core Django framework can be seen as an MVC architecture.[6] It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database ("**M**odel"), a system for processing HTTP requests with a web templating system ("**V**iew"), and a regular-expression-based URL dispatcher ("**C**ontroller").

Also included in the core framework are:

- a lightweight and standalone web server for development and testing
- a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database
- a template system that utilizes the concept of inheritance borrowed from object-oriented programming
- a caching framework that can use any of several cache methods

- support for middleware classes that can intervene at various stages of request processing and carry out custom functions
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals
- an internationalization system, including translations of Django's own components into a variety of languages
- a serialization system that can produce and read XML and/or JSON representations of Django model instances
- a system for extending the capabilities of the template engine
- an interface to Python's built-in unit test framework
- We can also create web apis with Django Rest Framework.

The main Django distribution also bundles a number of applications in its "contrib" package, including:

- an extensible authentication system
- the dynamic administrative interface
- tools for generating RSS and Atom syndication feeds
- a "Sites" framework that allows one Django installation to run multiple websites, each with their own content and applications.
- tools for generating Google Sitemaps.
- built-in mitigation for cross-site request forgery, cross-site scripting, SQL injection, password cracking and other typical web attacks, most of them turned on by default.
- a framework for creating GIS applications.

1.3 Introduction to NOSQL:

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

PostgreSQL is an enterprise-class open source database management system. It supports both SQL for relational and JSON for non-relational queries. It is backed by an experienced community of developers who have made tremendous contribution to make it highly reliable DBMS system.

PostgreSQL supports advanced data types and advance performance optimization, features only available in the expensive commercial database, like Oracle and SQL Server.

- Compatible with various platforms using all major languages and middleware
- It offers a most sophisticated locking mechanism
- Support for multi-version concurrency control
- Mature Server-Side Programming Functionality
- Compliant with the ANSI SQL standard
- Full support for client-server network architecture
- Log-based and trigger-based replication SSL
- Standby server and high availability
- Object-oriented and ANSI-SQL2008 compatible
- Support for JSON allows linking with other data stores like NoSQL which act as a federated hub for polyglot databases.

CHAPTER 2

EXISTING SYSTEM AND PROPOSED SYSTEM

2.1 EXISTING SYSTEM and its DISADVANTAGES:

2.1.1 Registrations :

Google Forms is a tool that allows collecting information from users through a personalized survey or quiz. The information is then collected and automatically connected to a spreadsheet. The spreadsheet is populated with the survey and quiz responses. The Forms service has undergone several updates over the years.

In this system event organizers in the college are using google form which are circulated among the watsapp groups for registration process. This system lacks security of the registration process and may not reach to all the participants.

2.1.2 Registration Fee Payment :

A screenshot of the transaction details will be uploaded in the google form as a png file which will be validated by organizers to confirm registration.

This system is not enabling validation properly. It increases human work to validate for the payment issues.

2.1.3 Tracking the Event :

All the updates regarding event are circulated through posters and watsapp group. Event registrations are tracked by the statistical data of the google form and needed to be stored somewhere so as to further assist the participants.

This system doesn't guarantee that information is being passed to each and every corner of the campus. It's a big headache for the organizers to get track of the registrations.

2.1.4 Marking Attendance of the Event:

Attendance should be marked manually by the organizers to ensure that the participant is physically present and to be noted and stored on papers or else digitally .

2.1.5 Issuing Certificates for the participants:

Issuing certificate involves a lot of human work . It includes designing a certificate template and then manually fill the certificate details and needed to be circulated among the participants . Organizers should ensure that the certificate has reached the participants successfully.

2.2 PROPOSED SYSTEM AND ITS ADVANTAGES:

- Event Tracker provides easy registration system for the students. All the students can register for their desired event with an easy click mechanism which routes them to payment gateway.
- Fee payment is securely integrated. A paytm payment gateway for web API is integrated so that registration process is transferred to the paytm . Payment is confirmed and validated by paytm and then statistics are send back to the Event Tracker's server.
- Event tracker helps us to track all the registered events any time in the students portal.
- Event Tracker comes in built with the attendance form which enables organizers to take attendance and store in the same database by user-friendly UI.
- Issuing certificates had become a big task for organizers. Event Tracker has a inbuilt mechanism to generate certificates for the participants and send them through the email and also allows users to download them from the portal anytime .

CHAPTER-3

PROJECT MANAGEMENT

3.1 PROJECT DEVELOPMENT APPROACH:

Mapping the right web application development process flow is a key to success for a project of any size. Despite the fact that the development of the web-based app is pretty similar to the regular desktop application one it has some differences that may impact the overall process and make it more complex in some aspects. In this article, we are going to explain each stage of the web application development process, name key specialists who can take part in the project, and map their area of responsibilities.

3.1.1 FIRST DISCUSSION

The first step aims to give an overview of the problem we want to be solved and/or our needs. If the needs aren't well understood it will lead to a failure in building a web app that aims to serve the purposes. The following questions have to be answered once this first step is done:

- What are the underlying needs behind the application to be built?
- What problems should the app solve?
- What needs do the application fill? What will be the impact on our business?
- How will the application be used?
- What could be the consequences of the delay in building the app for our business?

3.1.2 REQUIREMENTS ANALYSIS

In the previous step, the goal was to identify the objectives of web app implementation. The objectives can be complex, the business logic being often quite complex in general. It is then important at this stage that these complex objectives can be broken down into more manageable tasks and therefore easier to implement, validate, and test.
At the end of this step, the following tasks should have been completed:

- Reformulate the underlying needs and goals
- Think in terms of steps needed to complete the project
- Identify each different feature and module
- Break down every goal into simple tasks

- Consolidate the different analyses and approaches in a document so that validation can be carried out.

3.1.3 TIMELINE AND COST ESTIMATION

Experience is a good ally when it comes to making a good estimation. If a similar task has been done before, it becomes easier to give an accurate estimation and time tracking becomes our reference. From the previous step, we are able to have all the pieces we are going to put together from end to end. For each task, we analyze its complexity and do a classification.

- Perform a complexity analysis on each task
- Classify each task
- List all required development for each task: UI, logic, test.

3.1.4 THE DESIGN, CONCEPTION, AND PLANNING

The estimation gives us an idea of the time needed to complete the whole project. During this step, our goal is to answer the following questions: who is doing what and when? What is the best technical approach?

- Order each task in order to know which one should/must go before another
- Analyze dependency between all the tasks
- Identify every task that can be done in parallel
- Elaborate a timeline according to available resources or in case it is possible to have all the resources we need, identify the number of resources needed, and then plan the project
- Make an architectural choice.

Generally, at this stage, the financial and technical side of the web application creation is settled down and we have an idea of the budget.

3.1.5 THE DEVELOPMENT

This is the most well-known step. Once the product requirements document is established, the development phase can start. The goal of this phase is to create an application that meets

the needs identified in the previous steps, and as the needs generally evolve and new ideas of implementation may arise during the implementation phase, it is generally advised to use a methodology that allows flexibility and proactivity.

With each iteration in <u>Agile methodology</u>, we make sure that development is achieved towards the goals defined in the product requirements document.

The product requirements documentation should be used to implement all features, an efficient approach should be chosen in order to make sure the codes are behaving as expected by the specification document. TDD is one approach that aims to make sure that tests are dressed to test and simulate real behavior. Testing helps actors involved in the web application development process understand the code purposes and makes the hand out process easier. Good code coverage has proven to minimize bugs and is proof of the good design for easy post-production support and maintenance, easy system evolution.

3.1.6 THE TESTING PHASE

This phase is also called the <u>User Acceptance Test</u>. During that phase, a test group of users validates the set of features and report bugs if any, if there are changes to be made, they can be discussed with the project manager who will advise about the best way to take inputs and feedback into consideration.

3.1.7 THE DEPLOYMENT AND POST-PRODUCTION PHASE

Once every feature in a given release was validated and all the bugs reported fixed, the deployment phase can start, it may be a first launch or deployment of the new release, this is the moment when the web application will run in its real environment.

Some behaviours only appear in a production environment such as load balancer performance, it is important to allow times for support and maintenance in order to fix any bugs or performance issues that are discovered after the application went live.

New ideas for extension may appear once the application is up and running, the post-production phase is the best moment to talk about other needs.

The table below summarizes the lifecycle and expected output/delivery from each step.

WHO IS INVOLVED IN THE WEB APPLICATION DEVELOPMENT PROCESS FLOW?

From the web application development cycle we can identify the required participants in order to run this project smoothly. They are:

- The project manager: manage the project through its completion, mend the gap between the non-technical and the technical teams
- The product owner: the person who will be responsible to validate that the deliverables meet the organization's expectations and needs
- Back-end developer
- Front-end developer (UI/UX)
- Quality Assurance Tester
- System Administrator

3.2 SOFTWARE PROCESS MODEL – WATERFALL MODEL

To solve actual problems in an industry, software developer or a team of developers must incorporate a development strategy that encompasses the process, methods and tools layers and generic phases. This strategy is often referred to as process model or a software developing paradigm. A process model for software developing is chosen based on the nature of project and application, the methods and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem solving loop in which four distinct stages are encountered: Status quo, Problem definition, technical development and solution integration. Regardless of the process model that is chosen for a software project all of the stages coexist simultaneously at some level of detail.

Our Project Follows the Waterfall Model:

- **The Waterfall Model:**

    The steps of the typical Waterfall Model are:

    1. Requirement Definition.

2. System & Software Design.

3. Implementation.

4. Integration & System Testing.

5. Operation and Maintenance.

There have been some variations from the typical waterfall model for this project lifecycle.

They are:

1. Maintenance has been omitted from the current project.
2. Not all testing methods which are present in theoretical model are implemented.
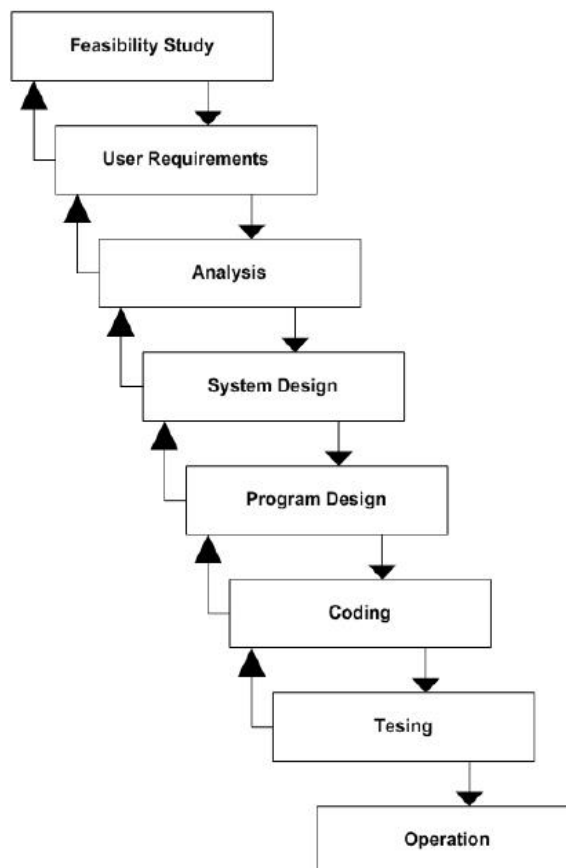
Fig. 3.2.1 Software Process model – Waterfall model

CHAPTER 4

FEASIBILITY STUDY

A feasibility study involves taking a judgment call on whether a project is doable. The two criteria to judge feasibility are **cost required** and **value to be** delivered. A well-designed study should offer a historical background of the business or project, a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, such studies precede technical development and project implementation

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions.

4.1 Technical Feasibility

Technical feasibility involves evaluation of the hardware and the software requirements of the proposed system.

It was easy to select the **Django** framework for our system. It has the ability for the needed security for the information that we are ingesting from various sources. It also allows for our responsive web application to be developed with ease using the combination of HTML, CSS, JavaScript and **Python**.

4.2 Economic Feasibility

Economic Feasibility helps in assessing the viability, cost, and benefits associated with projects before financial resources are allocated. This assessment typically involves a cost/ benefits analysis of the project.

The application is so designed that it requires minimal cost and eliminates costs as there would minimal need for manual work. The technologies used, help in understanding the user without any investment. As the machine will be trained it reduces the cost that is required to deploy the man power and also eliminates the problem of time consumption.

4.3 Legal Feasibility

The proposed system doesn't conflict with legal requirements like data protection acts or social media laws. It ensures legal data access and gives prominence to data security.

4.4 Operational Feasibility

The application involves design-dependent parameters such as reliability, maintainability, supportability, usability, disposability, sustainability, affordability, and others.

It minimizes the drawbacks of the current system by building an application that automatically resolves the user queries and helps to analyse the user data.

4.5 Scheduling Feasibility

The project development took place in a timely process by understanding time schedules of the project and maintaining a good time line for project development.

CHAPTER 5

SYSTEM REQUIREMENT STUDY

5.1 SOFTWARE REQUIREMENTS:

5.1.1. Python:

Python is   an interpreted, high-level, general-purpose   programming   language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Fig. 5.1.1.1 Python logo

5.1.2 Django :

Django is basically a high-level Python web application framework that enables the rapid development of web applications. It achieves so with pragmatic, much cleaner design

and is also easy to use (in comparison of other frameworks) thus is very popular among web developers.

It is a backend framework used to resolve problems of connectivity with databases, other server problems, **SEO solutions,** etc so that a web developer need not write the same code for the similar modules (like database connection, admin interface) for each website. All the functionality comes in the Django framework in the form of web applications. You just have to import those applications according to your need and thus you can concentrate more on the unique application of your website rather than dealing with all these backend problems.

Fig 5.1.2.1 django logo

Django can be installed easily using **pip(**Preferred installer program**)**.

In the command prompt, execute the following command: **pip install django**. This will download and install Django.

5.1.3 PostgreSQL

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

Fig. 5.1.3.1 PostgreSQL logo

5.1.4 Visual Studio Code

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code's source code comes from Microsoft's free and open-source software **VSCode** project released under the permissive Expat License,[8] but the compiled binaries are freeware for any use.

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents claiming to use it.

CHAPTER 6

SOFTWARE DESIGN

6.1 MVC

6.1.1   Definition

The **Model-View-Controller (MVC)** framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. Hence the abbreviation MVC. Each architecture component is built to handle specific development aspect of an application. MVC separates the business logic and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC architecture has become popular for designing web applications as well as mobile apps.

6.1.2   Features of MVC

- Easy and frictionless testability. Highly testable, extensible and pluggable framework
- Offers full control over your HTML as well as your URLs
- Leverage existing features provided by ASP.NET, JSP, Django, etc.
- Clear separation of logic: Model, View, Controller. Separation of application tasks viz. business logic, Ul logic, and input logic
- URL Routing for SEO Friendly URLs. Powerful URL- mapping for comprehensible and searchable URLs
- Supports for Test Driven Development (TDD)

6.1.3   MVC Architecture

Three important MVC the components are:

- Model: It includes all the data and its related logic
- View: Present data to the user or handles user interaction
- Controller: An interface between Model and View components

Let's see each other this component in detail:
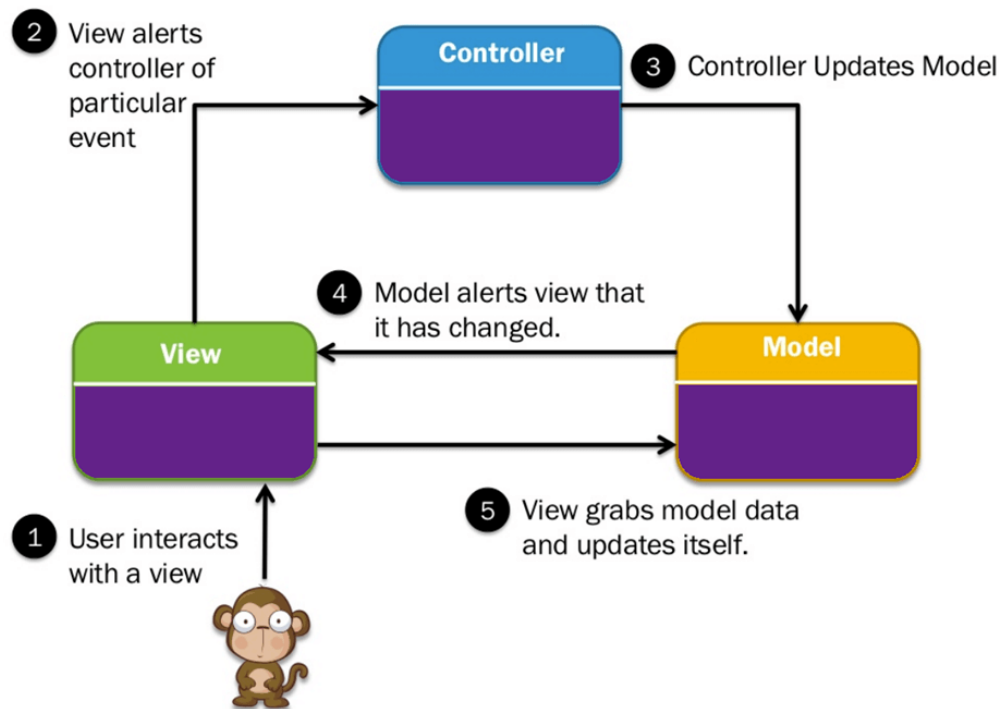
Fig. 6.1.3.1 MVC Architecture

6.1.3.1 View

A View is that part of the application that represents the presentation of data.

Views are created by the data collected from the model data. A view requests the model to give information so that it resents the output presentation to the user.

The view also represents the data from chats, diagrams, and table. For example, any customer view will include all the UI components like text boxes, drop downs, etc.

6.1.3.2 Controller

The Controller is that part of the application that handles the user interaction. The controller interprets the mouse and keyboard inputs from the user, informing model and the view to change as appropriate. A Controller send's commands to the model to update its state(E.g., Saving a specific document). The controller also sends commands to its associated view to change the view's presentation (For example scrolling a particular document).

6.1.3.3 Model

The model component stores data and its related logic. It represents data that is being transferred between controller components or any other related business logic. For example, a Controller object will retrieve the customer info from the database. It manipulates data and send back to the database or use it to render the same data.

It responds to the request from the views and also responds to instructions from the controller to update itself. It is also the lowest level of the pattern which is responsible for maintaining data.

6.1.4 Advantages of MVC: Key Benefits

Here, are major benefits of using MVC architecture.

- Easy code maintenance easy to extend and grow
- MVC Model component can be tested separately from the user
- Easier support for new type of clients
- Development of the various components can be performed parallelly.
- It helps you to avoid complexity by dividing an application into the three units. Model, view, and controller
- It only uses a Front Controller pattern which process web application requests through a single controller.
- Offers the best support for test-driven development
- It works well for Web apps which are supported by large teams of web designers and developers.
- Provides clean separation of concerns(SoC).
- Search Engine Optimization (SEO) Friendly.
- All classed and objects are independent of each other so that you can test them separately.
- MVC allows logical grouping of related actions on a controller together.

6.1.5 Disadvantages of using MVC

- Difficult to read, change, to unit test, and reuse this model
- The framework navigation can some time complex as it introduces new layers of abstraction which requires users to adapt to the decomposition criteria of MVC.
- No formal validation support.

- Increased complexity and Inefficiency of data
- The difficulty of using MVC with the modern user interface
- There is a need for multiple programmers to conduct parallel programming.
- Knowledge of multiple technologies is required.
- Maintenance of lots of codes in Controller.

6.2 Django's MVC :

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django's specificity for the Model-View-Template (MVT) pattern.

6.2.1 MVC Pattern

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. Check our MVC tutorial here to know more.

6.2.2 DJANGO MVC - MVT Pattern

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 following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request −
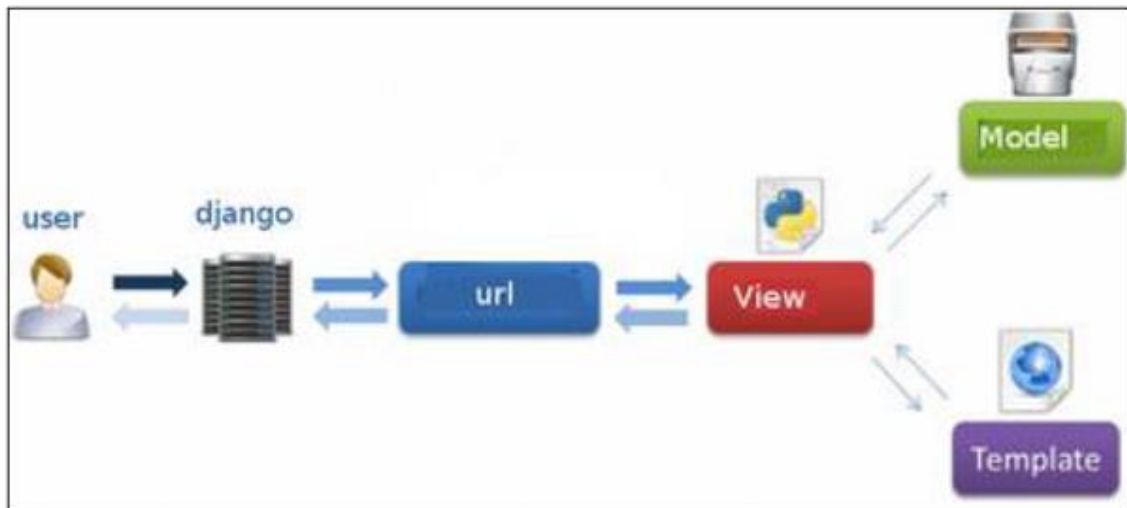
Fig. 6.2.2.1 DJANGO's MVC pattern

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

CHAPTER 7

FEATURES OF EVENT TRACKER

7.1 STUDENT REGISTRATION MODULE:

The main purpose of the Event Tracker is to provide all the functionality regarding Students registrations. So this module clearly depicts the students account creation to start with the app.

7.2 ORGANIZER REGISTRATION MODULE:

Organizer, being the data provider for the website will need to register n account in the application so as to add an event.

7.3 FUNCTIONALITIES OF STUDENTS:

- Can view the list of Events.
- Can view the details of particular events.
- Can enroll for the Event.
- Can View his/her own registered Events.
- Can track the event and get updated with the event information.
- Can Download Certificate once the participation is successful.
- Can submit feedback for the enrolled Events.

7.4 FUNCTIONALITIES OF ORGANIZERS:

- Can ADD the event into the application.
- Can UPDATE the event details whenever necessary.
- Can view all the registered students.
- Can track the registrations.
- Can mark the event attendance at the time of event.
- Can issue certificate for all the participants.
- Can send mail to the participants with certificate attached.
- Can send mail to the absentees.

CHAPTER 8

IMPLEMENTATION

8.1 DATABASE MODELS:

```python
class User(AbstractUser):
    is_student = models.BooleanField(default=False)
    is_organizer = models.BooleanField(default=False)



class Organizer(models.Model):
    name = models.CharField(max_length=30, verbose_name="N
ame")
    color = models.CharField(
        max_length=7,
        help_text='Specify color in Hexadecimal format. eg
 : #007f65 \n Colors in Hexa-
Decimal format : <a href = "https://htmlcolorcodes.com/">H
TML ColorPicker</a>',
        default="#007bff",
        verbose_name="Color of the badge",
    )


    description = models.TextField(
        max_length=1000,
        help_text="Enter a brief description of the Organi
sation Conducting the Event",
        verbose_name="Description",
    )


    email = models.EmailField()


    contact = PhoneNumberField(null=False, blank=False, un
ique=True)
```

```python
    def get_absolute_url(self):
        """Returns the url to access a particular author instance."""
        return reverse("organizer-detail", args=[str(self.id)])


    def __str__(self):
        return f"{self.name}"


    def get_html_badge(self):
        name = escape(self.name)
        color = escape(self.color)
        html = (
            '<span class="badge badge-primary" style="background-color: %s">%s</span>'
            % (color, name)
        )
        return mark_safe(html)



class Event(models.Model):
    owner = models.ForeignKey(User, on_delete=models.CASCADE, related_name="events")
    name = models.CharField(max_length=255, null=True, verbose_name="Name")
    description = RichTextUploadingField(blank=True, null=True)
    organizer = models.ForeignKey(
        Organizer, on_delete=models.SET_NULL, null=True, related_name="events"
    )
    dateofEvent = models.DateField(null=True, blank=True, verbose_name="Date of Event")
```

```python
    time = models.TimeField(null=True, blank=True, verbose
_name="Time of Event")
    registrationFee = models.DecimalField(
        max_digits=6, decimal_places=2, verbose_name="Regi
ster Amount"
    )
    img = models.ImageField(upload_to="images/", verbose_n
ame="Poster")


    def __str__(self):
        return self.name


    def get_absolute_url(self):
        return reverse("event-
detail", args=[str(self.id)])



class Student(models.Model):
    user = models.OneToOneField(
        User, on_delete=models.CASCADE, primary_key=True,
related_name="student"
    )
    events = models.ManyToManyField(Event, through="Enroll
edEvents")
    interests = models.ManyToManyField(Organizer, related_
name="interested_students")


    def __str__(self):
        return self.user.username



class EnrolledEvents(models.Model):
    student = models.ForeignKey(
```

```
        Student, on_delete=models.CASCADE, related_name="e
nrolled_events"
    )
    event = models.ForeignKey(
        Event, on_delete=models.CASCADE, related_name="enr
olled_events"
    )
    date = models.DateTimeField(auto_now_add=True)
    present = models.BooleanField(default=True)
    certificate = models.BinaryField(null=True, blank=True
)
    feedback = models.TextField(null=True, blank=True)
```

MODELS:

- User – Extends the User abstract model.
- Organizer – Stores the details of the organizer.
- Event : Stores the information regarding Event
- Student : Stores the student registration details.
- Enrolled Events : Stores the enrolment details of the student.

8.2 FRONT END

8.2.1 REST API

8.2.1.1 REST API KEY ELEMENTS

REST is a way to access resources which lie in a particular environment. For example, you could have a server that could be hosting important documents or pictures or videos. All of these are an example of resources. If a client, say a web browser needs any of these resources, it has to send a request to the server to access these resources. Now REST defines a way on how these resources can be accessed.

The key elements of a RESTful implementation are as follows:

- **Resources** – The first key element is the resource itself. Let assume that a web application on a server has records of several employees. The URL of the web application is http://localhost:5000 locally. To access data from table one can issue command http://localhost:5000 . The command tells web server to provide all rows in table.

- **Request Verbs** - These describe what you want to do with the resource. A browser issues a GET verb to instruct the endpoint it wants to get data. However, there are many other verbs available including things like POST, PUT, and DELETE. So in the case of example http://localhost:5000/Datahandler the web browser is actually issuing a GET Verb because it wants to get the details of the employee record.

- **Request Headers** – These are additional instructions sent with the request. These might define the type of response required or the authorization details.

- **Request Body** - Data is sent with the request. Data is normally sent in the request when a POST request is made to the REST web service. In a POST call, the client actually tells the web service that it wants to add a resource to the server. Hence, the request body would have the details of the resource which is required to be added to the server. A user can add row to the table using form which sends data ,the data sent is in json format.

- **Response Body** – This is the main body of the response. So in our example, if we were to query the web server via the request , http://localhost:5000/Datahandler the web server returns a json object with all the required rows of table in the Response Body.

- **Response Status codes** – These codes are the general codes which are returned along with the response from the web server. An example is the code 200 which is normally returned if there is no error when returning a response to the client.When the inserted image in row can't be processed status code 422 is returned in json body along with error message.

8.2.1.2 Restful Methods

We have a RESTful tweb service is defined at the location, http://localhost:5000/Datahandler . When the client makes any request to this web

service, it can specify any of the normal HTTP verbs of GET, POST, DELETE and PUT. Below is what would happen If the respective verbs were sent by the client.

1. POST – This would be used to create a new row using the RESTful web service.
2. GET - This would be used to get a list of all rows using the RESTful web service.
3. DELETE - This would be used to delete a row using the RESTful web service.

8.2.2 VIEW FUNCTIONS:

8.2.2.1 ATTENDANCE MARKING VIEW:

```python
@login_required
@organizer_required
def save_data(request, pk):
    if request.method == "POST":
        # print(request.POST.getlist("present"))
        student_list = request.POST.getlist("present")
        event = Event.objects.get(pk=pk)
        # print(event)

        EnrolledEvents.objects.filter(event=event).filter(
            student__in=student_list
        ).update(present=True)

        EnrolledEvents.objects.filter(event=event).exclude(
            student__in=student_list
        ).update(present=False)
        messages.success(request, "Attendance was Updated Successfully! Go Ahead!")
        return redirect("organizers:event_change_list")
```

A "POST" request is generated by the API so that the data in the data base is updated accordingly.

8.2.2.2 GENERATING CERTIFICATE VIEW:

```python
@login_required
@organizer_required
def generate_certificate(request, pk):
    event = Event.objects.get(pk=pk)
    if request.method == "POST":
        enrolled_students = event.enrolled_events.select_related(
            "student__user"
        ).filter(present=True)
        img_url = os.path.join(settings.MEDIA_ROOT, "certi_template.jpg")
        font_path = os.path.join(
            os.path.join(settings.BASE_DIR, "static"), "fonts/Pacifico.ttf"
        )
        font_path1 = os.path.join(
            os.path.join(settings.BASE_DIR, "static"), "fonts/Alexandroupoli.ttf"
        )
        for s in enrolled_students:
            name = s.student.user.first_name + " " + s.student.user.last_name
            text_y_position1 = 550
            text_y_position2 = 780
            text_y_position3 = 950
            img = Image.open(img_url, mode="r")
            image_width = img.width
            image_height = img.height
            draw = ImageDraw.Draw(img)
            date = str(event.dateofEvent)
            font1 = ImageFont.truetype(font_path, 100)
            font2 = ImageFont.truetype(font_path, 70)
            font3 = ImageFont.truetype(font_path, 60)
            font4 = ImageFont.truetype(font_path1, 60)
            text_width1, _ = draw.textsize(name, font=font1)
            text_width2, _ = draw.textsize(event.name, font=font2)
            draw.text(
```

```python
            ((image_width - text_width1) / 2, text_y_position1,),
            name,
            font=font1,
            fill=(96, 96, 96),
        )
        draw.text(
            ((image_width - text_width2) / 2, text_y_position2,),
            event.name,
            font=font2,
            fill=(96, 96, 96),
        )
        draw.text(
            (1200, text_y_position3,), date, font=font3, fill=(96, 96, 96),
        )
        draw.text(
            (570, text_y_position3,), "Vnrvjiet", font=font4, fill=(96, 96, 96),
        )
        # img.show()
        file_path = os.path.join(settings.MEDIA_ROOT, "gmail_certi/certificate.jpg")
        img.save(file_path)
        student_instance = EnrolledEvents.objects.filter(event=event).filter(
            student=s.student
        )
        student_instance.update(certificate=img.tobytes())
        subject = "Certificate of " + event.name + " from Terrific Trio!"
        message = "Congratulations!!! You have succesfully achieved Certificate from
Terrific Trio.Please download it from the website!"
        recepient = s.student.user.email
msg = EmailMessage(subject, message, settings.EMAIL_HOST_USER, [recepient])
        msg.content_subtype = "html"
        msg.attach_file(file_path)
        msg.send()

    return redirect("organizers:event_change_list")
```

8.2.2.3 MAILING ABSENTEES VIEW:

```python
@login_required
@organizer_required
def absent_mail(request, pk):
    event = Event.objects.get(pk=pk)
    if request.method == "POST":
        enrolled_students = event.enrolled_events.select_related(
            "student__user"
        ).filter(present=False)
        for s in enrolled_students:
            subject = event.name
            message = (
                "Hey "
                + s.student.user.first_name
                + " "
                + s.student.user.last_name
                + "! We are sorry to inform that you are not eligible to receive certificate as you didn't turn up at the event on dated!"
            )
            recepient = s.student.user.email
            msg = EmailMessage(subject, message, settings.EMAIL_HOST_USER, [recepient])
            msg.send()
        return redirect("organizers:event_change_list")
```

8.2.2.4 DOWNLOAD CERTIFICATE VIEW:

```python
@login_required
@student_required
def download_certi(request, pk):
    event = Event.objects.get(pk=pk)
    student = request.user.student
    img_url = os.path.join(settings.MEDIA_ROOT, "certi_template.jpg")
    img = Image.open(img_url, mode="r")
```

```python
    image_width = img.width
    image_height = img.height
    if request.method == "GET":
        bin_img = EnrolledEvents.objects.get(event=event, student=student).certificate
        image = Image.frombytes("RGB", (image_width, image_height), bin_img, "raw")
        # print(image)
        file_path = os.path.join(settings.MEDIA_ROOT, "certi.jpg")
        image.save(file_path)
        if os.path.exists(file_path):
            with open(file_path, "rb") as fh:
                response = HttpResponse(
                    fh.read(), content_type="application/vnd.ms-excel"
                )
                response["Content-Disposition"] = (
                    "inline; filename="
                    + os.path.basename(
                        event.name + "_" + request.user.first_name + ".jpg"
                    )
                )
                return response
        raise Http404
```

CHAPTER 9

USER INTERFACE

User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI refers more to the appearance or presentation of a website than the way someone might use it, including colors, layout, typography, and other design elements.
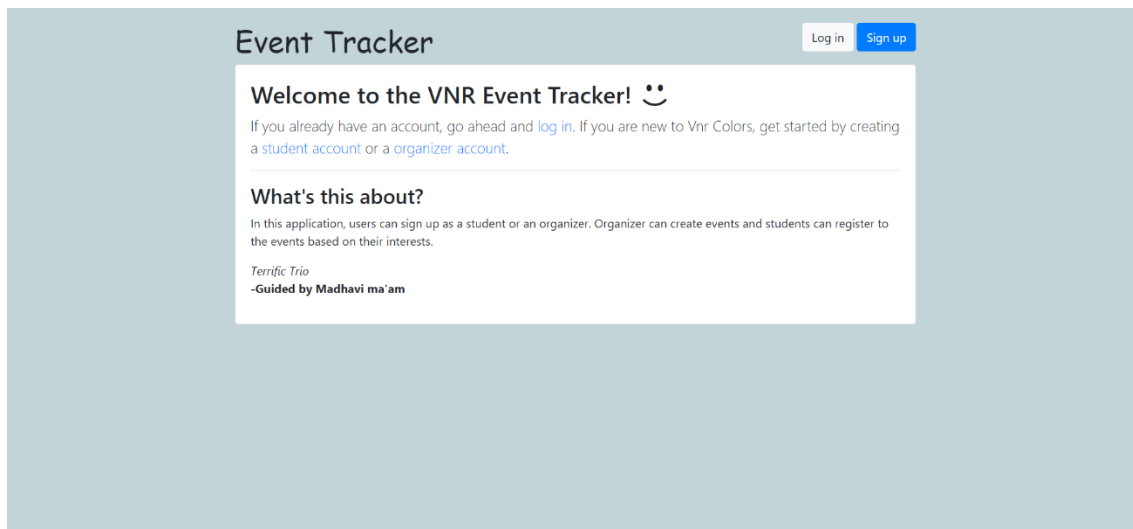
Fig 9.1 Home page of Website

9.1 User Validation Forms

Fig. 9.1.1 Login page

Fig 9.1.2 User Signup view

Fig 9.1.3 Student Signup Form

- *User need to create account so as to access the application.*
- *Users can be either Student or a Organizer.*
- *Registered users can login into the application with their credentials.*
- *Student will need to provide personal information for further registration processes.*

9.2 Organizer View
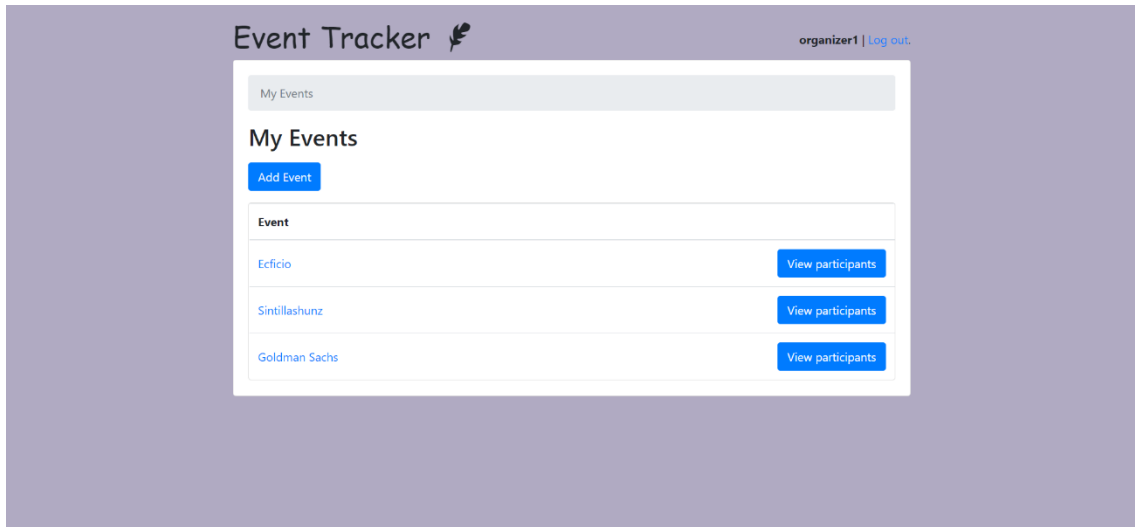
Fig 9.2.1 Organizer's Event List

After successful registration organizer will be directed to his home page.

- This page is a List View page which displays list of all the events organized by the particular organizer.

- Organizer will be able to add a new event using Add Event button.

- Organizer is able to view the details of his enrolled participants by accessing View Participants button.

- Details of the event can be modified or revisited by clicking on the event name.
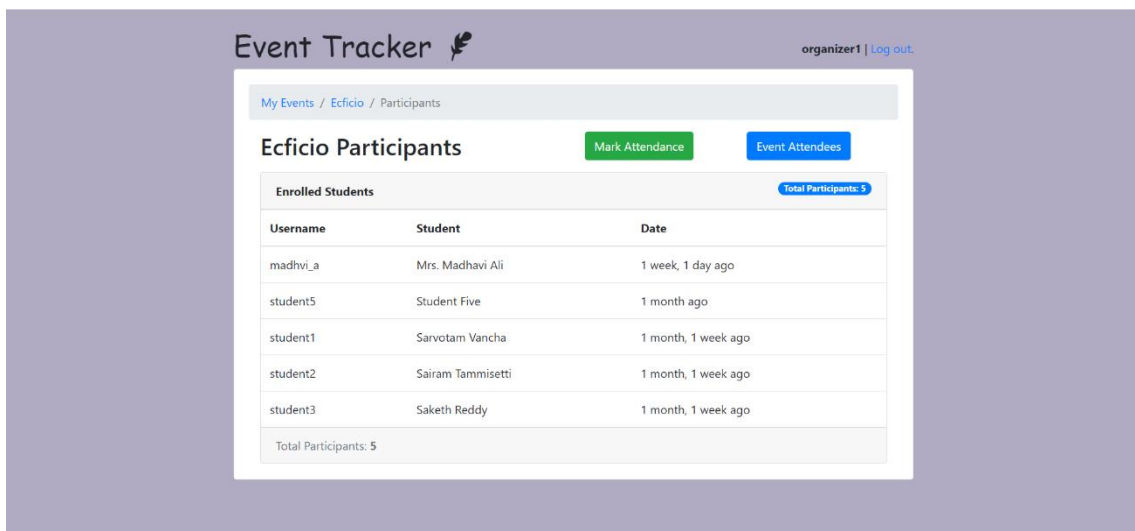
Fig 9.2.2 Participant's List

Here comes the participants forum:

- Organizer will be able to view the details of his enrolled students.
- He will be able to mark attendance for the participants of the event by clicking the Mark Attendance button.
- He can view the event statistics by pressing Event Attendees button.
- This page is dynamically updated and the registered participants will be added to the list.
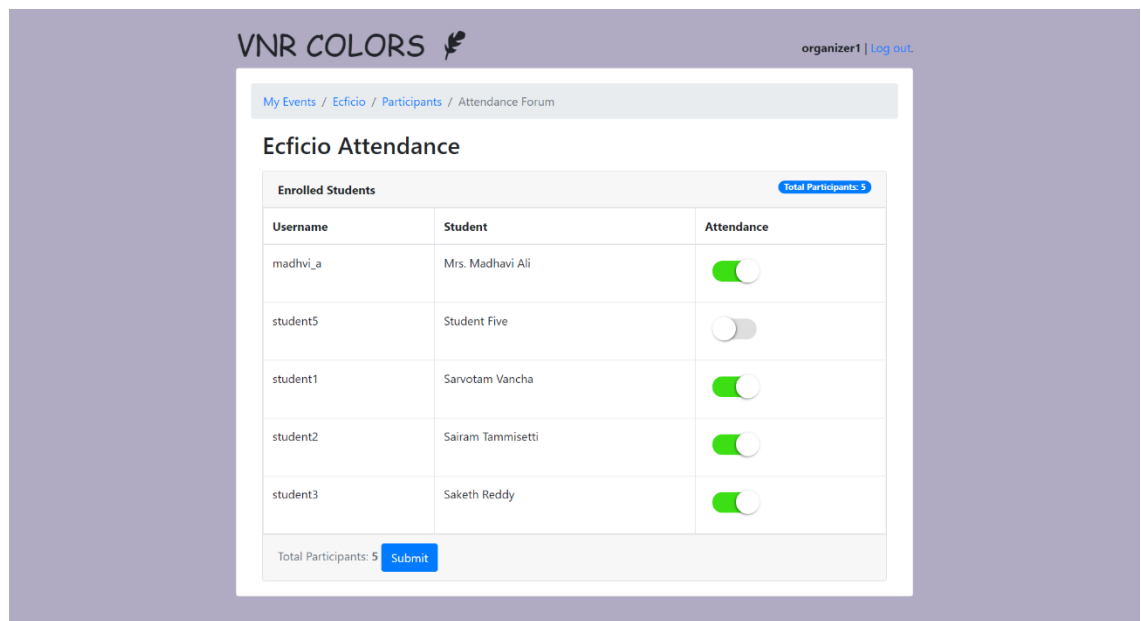
Fig 9.2.3 Attendance Forum

Attendance Forum :

- Organizer will be able to mark attendance for the enrolled students and categorize them as presentees and absentees.
- On submit the attendance will be updated in the database.
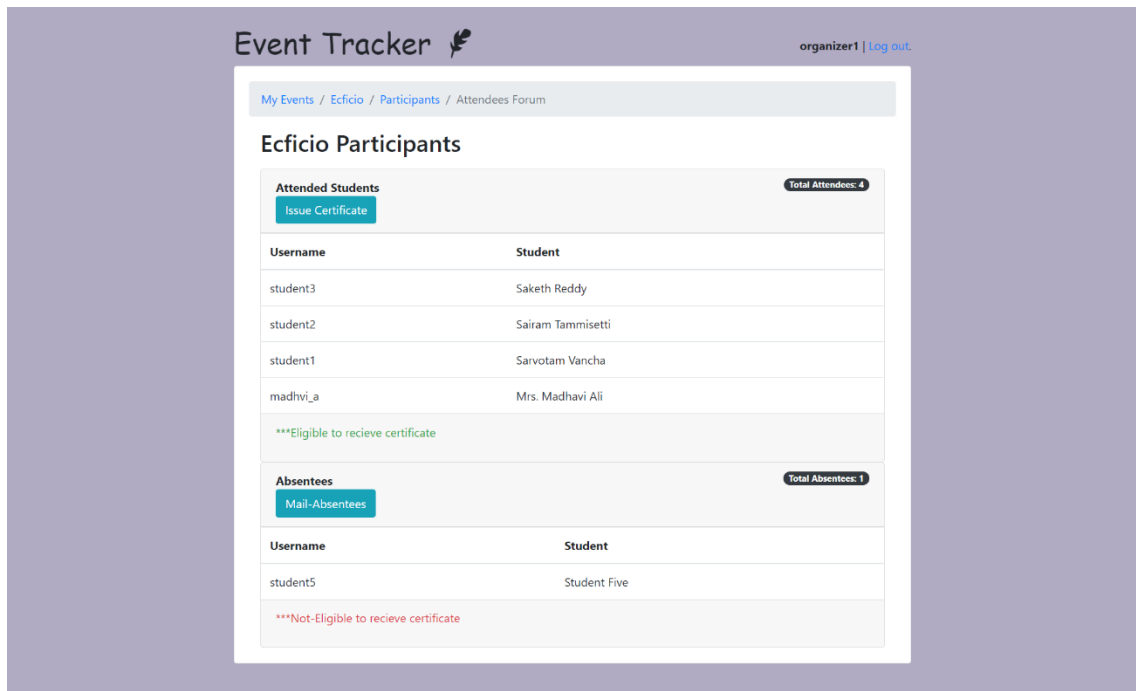
Fig 9.2.4 Attendees Forum

Attendees Forum:

- Here organizer will be able to view the number of students present and absent for the event.
- Organizer will be able to generate and issue certificate for the participants by clicking Issue Certificate button.
- Organizer will also be able to mail absentees for further assistance by clicking mail absentees button.

Fig 9.2.5 Event Create Form

9.3 Student View:

Fig 9.3.1 Student's Events List

Students Home View:

- Students will be able to view the list of all events uploaded by various organizers.
- They can access the details of the event by clicking Details button.
- Pagination is provided at the bottom left of the page.
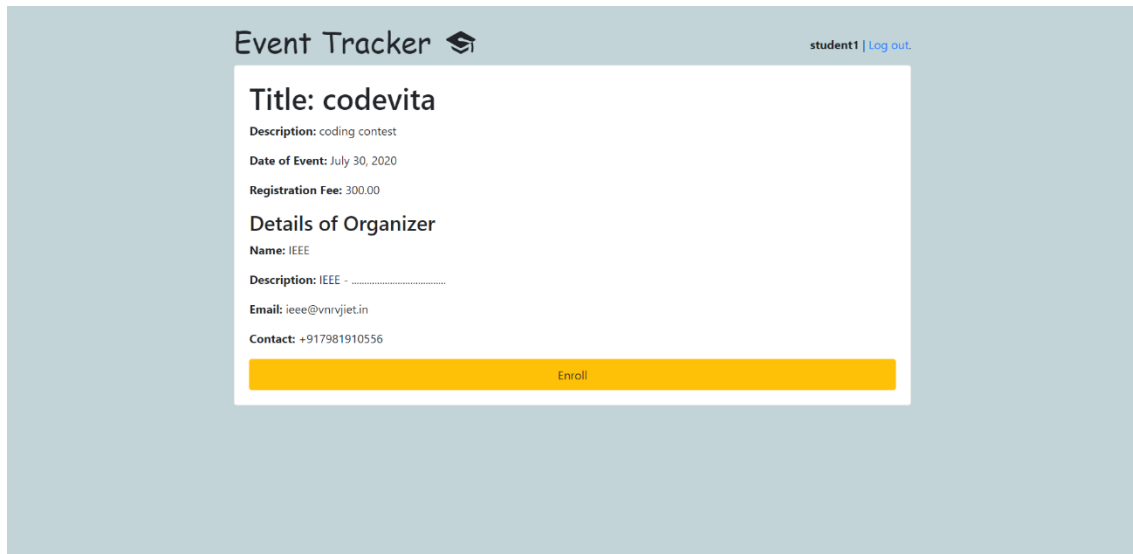- Searching is made easier with the default search enabled.

Fig 9.3.2 Event Detail

Event Detail view :

- This views provides information regarding the particular event.
- This is the view where the event enrolment button is provided.
- On successful verification one can enroll for the event by pressing enroll button.
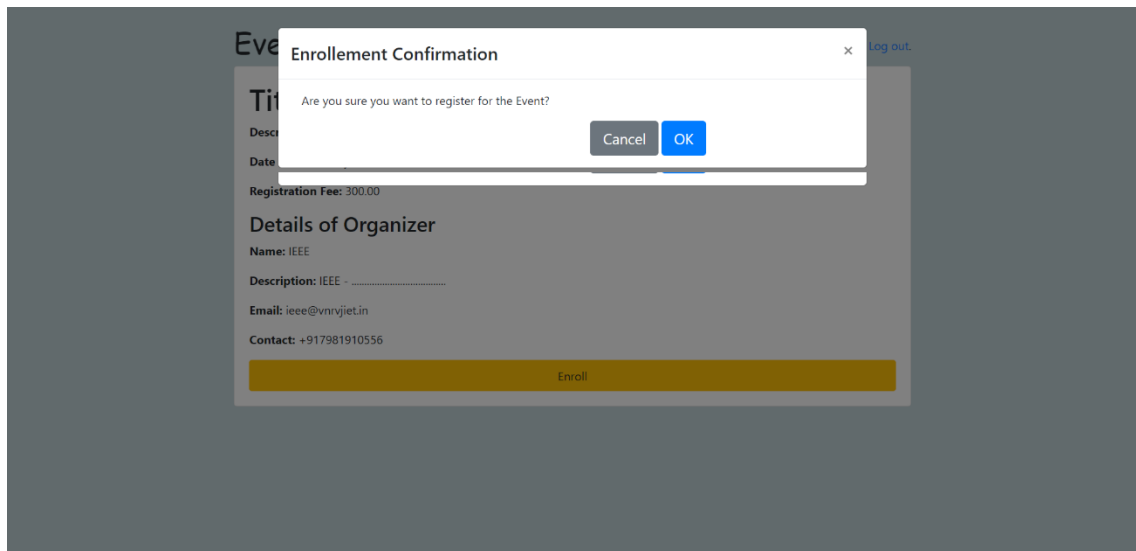
Fig 9.3.3 Event Enrolment Confirmation

A confirmation modal is popped on click of enroll event button.
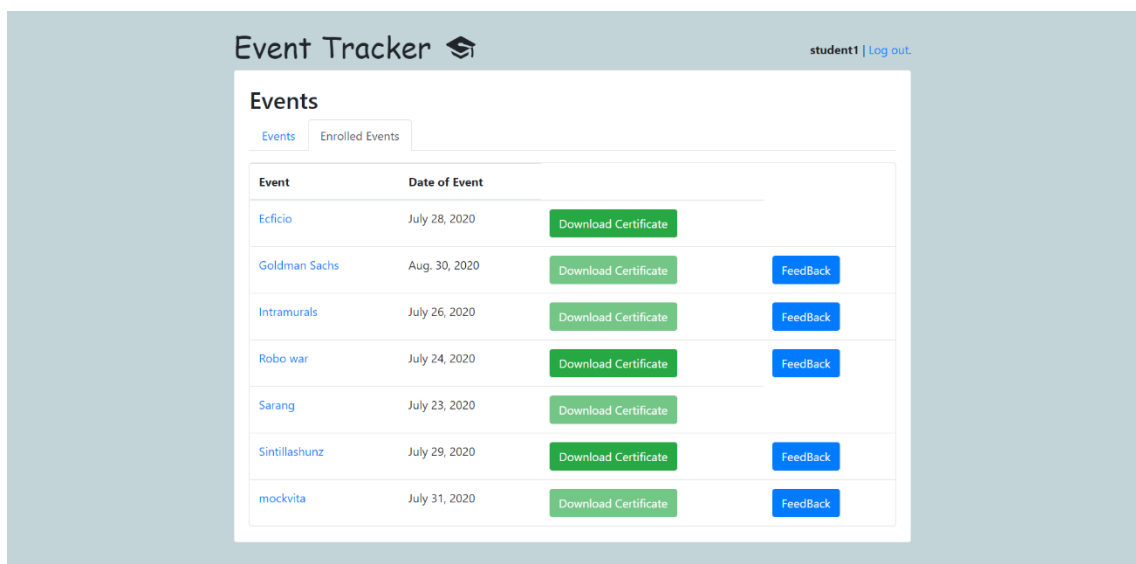
Fig 9.3.4 Enrolled Events list

- Student will be able to vie the list of his own registered events.
- Student can download the certificate once it is issued by the organizer.
- Students will also be asked to submit their feedback of the registered events.

9.4 Mail View

fig 9.4.1 Event participation success mail

Once the certificate is issued by the organizer then it will be seen the student mail box.

Fig 9.4.2 Automatically generated certificate of participation

Here is the certificate generated by PIL .Template of the certificate needed to be uploaded by admin.

Fig 9.4.3 Event Participation Failure mail

Absentees will receive the following mail enquiring about their cause of absence.

CHAPTER 10

FUTURE SCOPE

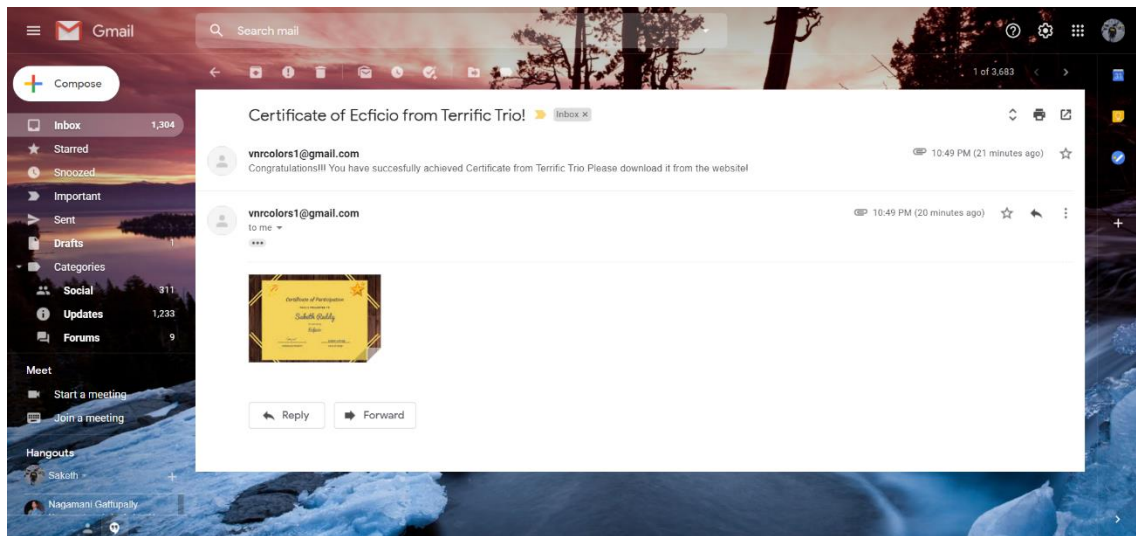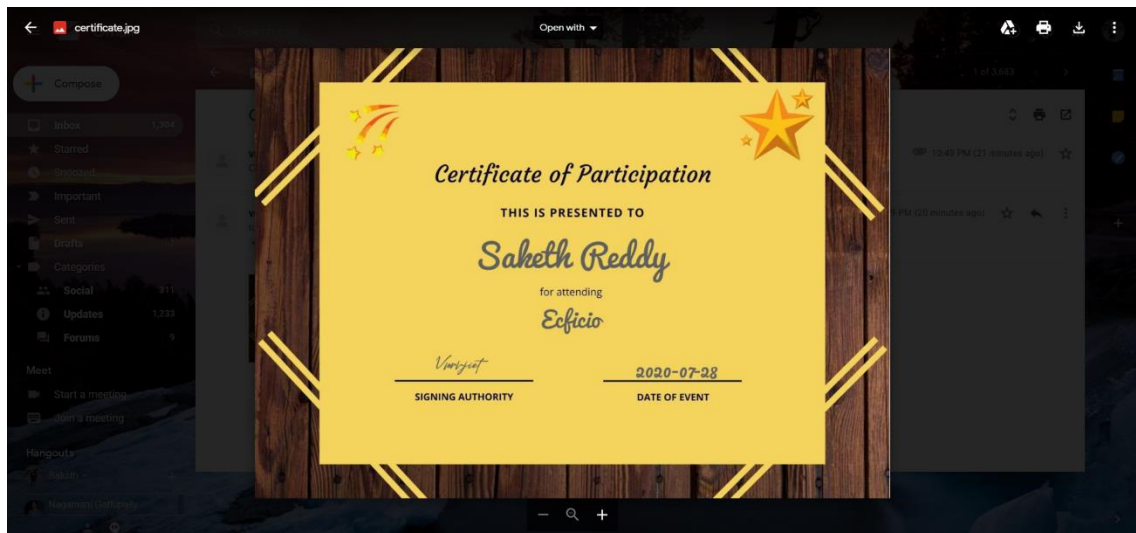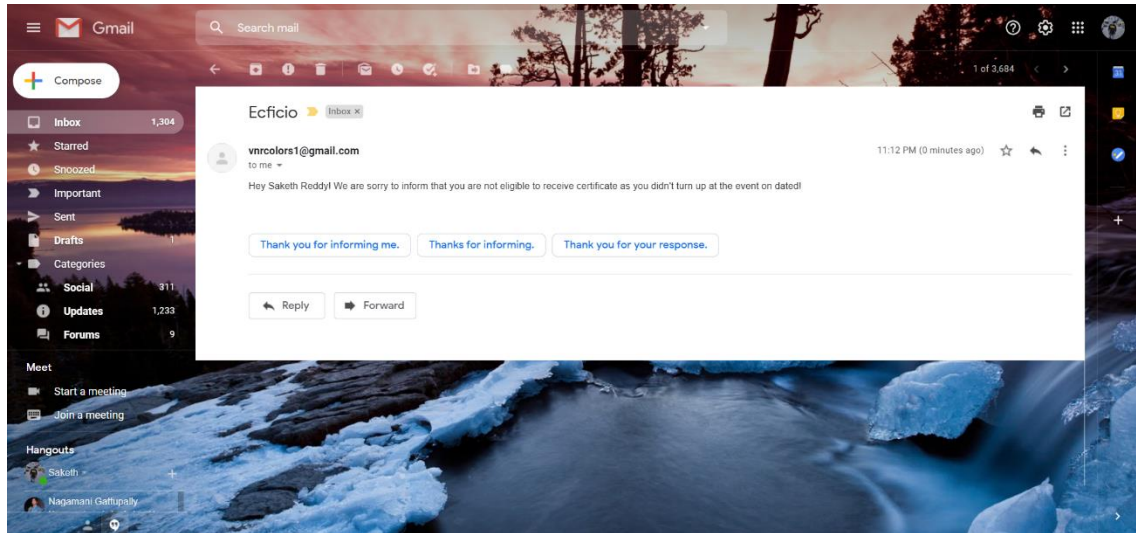Event is organized in almost all the colleges but all the work done is manually. The events are decided and then asked to students whether they will coordinate or not then all the details are maintained in files manually. Many colleges have websites for events that they organize but this website is used for displaying the various events details that are going to be conduct in those colleges for informing the other college students. Students also not get the proper platform for finding events which helps to build their resume. There is no such android application or web application for Event management or organization by which automatic record will maintain, also show all events to participants by which they can apply online and event can be organized.

Our product is a communication and management application which will useful for event organization purpose. Our product will be used to communication purpose, event timing scheduling etc. It is a college level application which will fulfill the communication, effective management and timetable scheduling requirements of the organizers and students coordinators of the particular event, online registration of the participants. It is not meant for general social networking use. However, it will not provide all the facilities and services provided by social networking applications like facebook and whatsapp but provide enough functionality like messaging, sending files or reports regarding events like poster of event or demo of certificate, payment receipt etc., creating and managing groups for campaigning or for assigning different tasks, modification in schedule, notifying the changes in schedule etc.

This app is also designed to take attendance for the event participants. Further announcements will be send to students by mails. Attendees will receive certificates through mail and absents will receive a mail .

CHAPTER 11

CONCLUSION

So, basically main aim of proposed system is to develop an Web application which can be useful for events organizing and event publicity purpose. The proposed system will be useful for communicating with staff coordinators and students coordinators, conveying important messages to participants, exchanging important reports, bills, queries, etc. and participant students can view and register for events online. Not only that they can download certificates any time and they can view their participation in events.

We have prepared new system after identifying issues in existing manual system. In which easy to use GUI is proposed by which Student coordinators, Staff coordinator can view all the records which are necessary. Participant students can view and register for events online. The record maintenance are use of previous records becomes easy and effective communication between staff coordinators and student coordinators. Thus we will implement Event Tracker system to address the problem faced by event organizers with respect to communication and working methods.

The main goals of the project are:

- To reduce the communication gap between students / participants and event coordinators of the events.
- Online registration / application of participants. -Proper utilization of the schedule and digital resources.
- Easy to assign students for campaigning.
- To reduce the overhead of the organizing committee.
- To generate and send certificates online reducing lot of paper work as well as manual work.

The proposed system is dedicated for the Events Organizing purpose. The applications of the proposed system is,

- To communicate/messaging between event organizer and participants formally. -To notify the changes done in the event schedule.
- To exchange the data between students and staff coordinators. –To display all events to participant and they can apply online.
- To convey any important message to students coordinators or participants.

REFERENCES

WEBSITES

- **https://cloudinary.com/documentation/django_image_manipulation#image_optimizations**

- **https://www.geeksforgeeks.org/create-certificates-using-python-pil/**

- **https://djangoschools.herokuapp.com/**

- **https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html**

- **http://django-todo.org/todo/**

- **https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html**

- **https://django-sis.readthedocs.io/en/latest/**

- **https://www.geeksforgeeks.org/python-sessions-framework-using-django/**

- **https://www.tutorialspoint.com/django/django_sessions.htm**

- **https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website**

- **https://stormy-cove-10463.herokuapp.com/catalog/**

- **https://studygyaan.com/django/best-practice-to-structure-django-project-directories-and-files**

- **https://cloudinary.com/documentation/django_image_manipulation#image_optimizations**

JOURNALS

- Review on College Event Organizer - International Research Journal of Engineering and Technology (IRJET)- Volume: 04 Issue: 3 | Mar -2017 - https://www.irjet.net/archives/V4/i3/IRJET-V4I3121.pdf

Books

- **The definitive guide to** Django**: Web development done right https://books.google.com/books?hl=en&lr=&id=h2tR8p-4a9QC&oi=fnd&pg=PR27&dq=django&ots=Voqaj0cMMb&sig=Lw785f3qhHWEUjnwFLAVOjlXx5w**

- **Python web development with** Django
  https://books.google.com/books?hl=en&lr=&id=M2D5nnYlmZoC&oi=fnd&pg=PT31&dq=django&ots=vY-LKoeQKU&sig=U8H0yDmg3Frr5uNuMHz3fKZPPD0

- **Practical Django Projects**
  https://books.google.com/books?hl=en&lr=&id=qfp4BAfAvVkC&oi=fnd&pg=PR11&dq=django&ots=_0kEPcZZZd&sig=weYaaHqwQQPhNVD_FyHqqKuqULo