**“TITLE OF THE PROJECT”**

***A Minor Project Report submitted to the***

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL**

***in partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

***Submitted by***

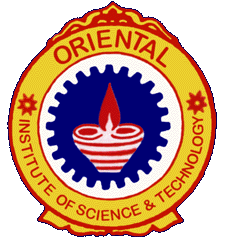
**ABC (0105IT000000)**

**EFJ (0105IT000001)**

**IJK (0105IT000002)**

***Under the Guidance of***

**Prof. XYZ**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**ORIENTAL INSTITUTE OF SCIENCE & TECHNOLOGY**

**BHOPAL (M.P.)-462021, INDIA**

**2018-19**

|  |  |
| --- | --- |
|  |  |

**ORIENTAL INSTITUTE OF SCIENCE & TECHNOLOGY**

Approved by AICTE, New Delhi & Govt. of M.P. Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

Oriental Campus, Thakral Nagar, Raisen Road, Bhopal-462 021 (M.P.) INDIA

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**BONAFIDE CERTIFICATE**

Certified that this project report **“……….TITLE OF THE PROJECT …………………”** is a bonafide work of “**…………..NAME OF THE CANDIDATE(S).…………”** who carried out the project work under my supervision.

Date: \_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_

**Prof. Pratik Buchke** <<Name>>

**HEAD OF THE DEPARTMENT SUPERVISOR** <<Academic Designation>>

Information Technology Information Technology

Oriental Institute of Science & Technology Oriental Institute of Science & Technology

Oriental campus, Raisen road, Bhopal-462021 Oriental campus, Raisen road, Bhopal-462021

**ACKNOWLEDGEMENT**

We wish to express our profound gratitude to **Prof. < GUIDE>, Department of Information Technology, Oriental Institute of Science and Technology Bhopal** for <his/her> invaluable guidance, sustained help and inspiration in carrying out the work.

We sincerely thank **Prof. Pratik Buchke, Head of** **Information Technology** Department, OIST Bhopal for his co-operation and guidance during the project work.

We wish to express our deep gratitude to **Dr. K.G. Sudhakar, Director, OIST Bhopal** for her kind permission to carry out the work at OIST Bhopal .

Finally we would like to thank all the faculty members and other persons who directly or indirectly helped us in carrying out this project work.

**ABC (0105IT000000)**

**EFJ (0105IT000001)**

**IJK (0105IT000002)**

**ABSTRACT**

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **Bonafide Certificate……………………………………………………………………………** | **i** |
| **Acknowledgements …………………………………………………………………………….** | **ii** |
| **Abstract ………………………………………………………………………………………..** | **iii** |
| **List of Figures …………………………………………………………………………………** | **vi** |
| **List of Tables …………………………………………………………………………..............** | **vii** |
| **List of Symbols…………………………………………………………………………………** | **viii** |
| **List of Abbreviations…………………………………………………………………………..** | **ix** |

|  |  |  |
| --- | --- | --- |
| **1.** | **introduction…………………………………………………………** |  |
| 1.1 | Objective……………………………………………………………………………….... |  |
| 1.2 | Scope……………………………………………………………………………….......... |  |
| 1.3 | Problem identification ………………………………………………………………….. |  |
| 1.4 | Methodology……………………………………………………………………………. |  |
| **2.** | **BACKGROUND AND LITERATURE SURVEY……………………** |  |
| 2.1 | Existing system…………………………………………………………………………... |  |
| 2.2 | Requirement Specifications…………………………………………………… |  |
| 2.3 | Feasibility Study……………………………………………………………………….... |  |
| 2.4 | Innovativeness and Usefulness………………………………………………………….. |  |
| 2.5 | Modification and improvement over the existing implementation………………............ |  |
| **3.** | **PROCESS MODEL….…………………………………………………..** |  |
| 3.1 | Software Process Model Used………………………………………………………….. |  |
| 3.2 | Proposed Project Model…………………………………………………………………. |  |
| 3.3 | Market Potential and Competitive Advantage …………………………………………. |  |
| 3.4 | Project Estimation……………………………………………………………………….. |  |
| 3.5 | Project Schedule………………………………………………………………………… |  |
| **4.** | **DESIGN………………………………………………..…………………** |  |
| 4.1 | Data Flow Diagram…………………………………………..…………………………. |  |
| 4.2 | Entity Relationship Diagram……………………………………………………………. |  |
| 4.3 | Flow Chart ……………………………………………………………………………… |  |
| 4.4 | USE-CASE diagrams…………………………………………………………………… |  |
| 4.5 | Sequence Diagram……………………………………………………………………… |  |
| 4.6 | Activity Diagram……………………………………………………………………….. |  |
| 4.7 | Class Diagram………………………………………………………………………….. |  |
| 4.8 | Algorithm………………………………………………………………………………. |  |
| **5.** | **TECHNOLOGY AND TOOLS USED…………………………………** |  |
| 5.1 | Front-end Tools (With brief description) ………………….............................................. |  |
| 5.2 | Back-end Tools (With brief description)………………………………………………… |  |
| 5.3 | Operating System (With brief description)……………………………………………… |  |
| 5.4 | Additional Software requirements (Web browser, utility tools etc.)……………………. |  |
| **6.** | **HARDWARE REQUIREMENT…………………………………………** |  |
| 6.1 | System Configuration (CPU, RAM, HDD, Network used)…………………………….. |  |
| **7.** | **CODING………………………………………………………………….** |  |
| **8.** | **TESTING …..…………………………………………………………….** |  |
| 8.1 | Testing Method Used …………………………………………………………………… |  |
| 8.2 | Test case with results ……………………………………………………………………. |  |
| 8.3 | Results analysis …………………………………………………………………………. |  |
| **9.** | **Screen Layouts……………………………………………………………** |  |
| **10.** | **Future Enhancement…………………………………………………….** |  |
| **11.** | **Appendices ………………………………………………………………..** |  |
| **12.** | **References** |  |

# 

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1.1 | System Model…………………….............................................................. |  |
| 1.2 | ER Diagram………………………………………………………………... |  |
| 1.3 | Use Case Diagram………………………………………………………… |  |
| 2.1 | Data Flow Diagram……………………………………………………….. |  |
| 3,2 | User Screen………………………………………………………………... |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 1.1 | XYZ……………………………………………………………………….. |  |
| 1.2 | XYZ……………………………………………………………………….. |  |
| 1.3 | XYZ……………………………………………………………………….. |  |
| 2.1 | XYZ……………………………………………………………………….. |  |
| 3,2 | XYZ……………………………………………………………………….. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ALGORITHMS**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 1.1 | XYZ……………………………………………………………………….. |  |
| 1.2 | XYZ……………………………………………………………………….. |  |
| 1.3 | XYZ……………………………………………………………………….. |  |
| 2.1 | XYZ……………………………………………………………………….. |  |
| 3,2 | XYZ……………………………………………………………………….. |  |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **DESCRIPTION** |
| CPU | Central Processing Unit |
| IP | Internet Protocols |
| TCP | Transmission Control Protocols |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**1. Project**

**1.1Introduction**

This project report will introductions how to built part of an engineering Notes using the Django framework.Django is an open source web application frame work,which is written in python.

This engineering Notes website built using Django and its has four major component each of which has different functionality but similar architecture .

In the project report I will demonstrate details of using Django to built one major component of this system ,the group component ,which is my major contribution to the whole system .Also the tech. and process which is showed here can be applied to build the other three components in the engineering Notes website as well as other complex database -drive website. Python Programming is intended for software engineers, systems analysts, program  
managers and user support personnel who wish to learn the Python programming language.Python is a general-purpose programming language that ibecoming more and more popular for doing data science. Companies worldwide are using Python to harvest insights from their data and get a competitive edge. Unlike any other Python tutorial, this course focuses on Python specifically for data science. In our Intro to Python class, you will learn about powerful ways to store and manipulate data as well as cool data science tools to start your own analyses. Enter DataCamp’s online Python curriculum.You may have heard of Python before, especially if you have been coding for a while.

If not, Python is a high level, general purpose programming language. What this means is that you can use it to code up anything from a simple game to a website supporting millions of users per month.

In fact, several high profile sites with millions of visitors per month rely on Python for some of their services. Examples include [YouTube](https://www.youtube.com/) and [Dropbox](https://www.dropbox.com/)

**1.2 Objectives**

Web design presents a challenge few have mastered. We have all used web sites that provide us with what we are looking for, and many more that don’t, but what makes some sites more appealing than others?

This tutorial explores the importance of having clearly defined and prioritised objectives when developing web sites.

So what is the purpose, or objectives of a web site? Primarily, to help customers, or other stakeholders. Ask yourself “How can my web site help my customers?”

* Help them buy something they need.
* Help them find information.
* Help them to save money and time.
* Help them to talk to the organisation.

The learning objectives of this course are:

* To understand why Python is a useful scripting language for developers.
* To learn how to design and program Python applications.
* To learn how to use lists, tuples, and dictionaries in Python programs.
* To learn how to identify Python object types.
* To learn how to use indexing and slicing to access data in Python programs.
* To define the structure and components of a Python program.
* To learn how to write loops and decision statements in Python.
* To learn how to write functions and pass arguments in Python.
* To learn how to build and package Python modules for reusability.
* To learn how to read and write files in Python.
* To learn how to design object‐oriented programs with Python classes.
* To learn how to use class inheritance in Python for reusability.
* To learn how to use exception handling in Python applications for error handling.

# 1.3 The scope of Django and Python in web development

The state of web-development is in a constant flux and you can never surely say when a game changer technology would come and change the current trends entirely, so it would be wise to take a peek into any new technology that is gaining enough momentum and stay updated with the current trends.

But, having said that, I don’t think python and django are going anywhere in the next 10–15 years. Let’s talk facts:*Python:*

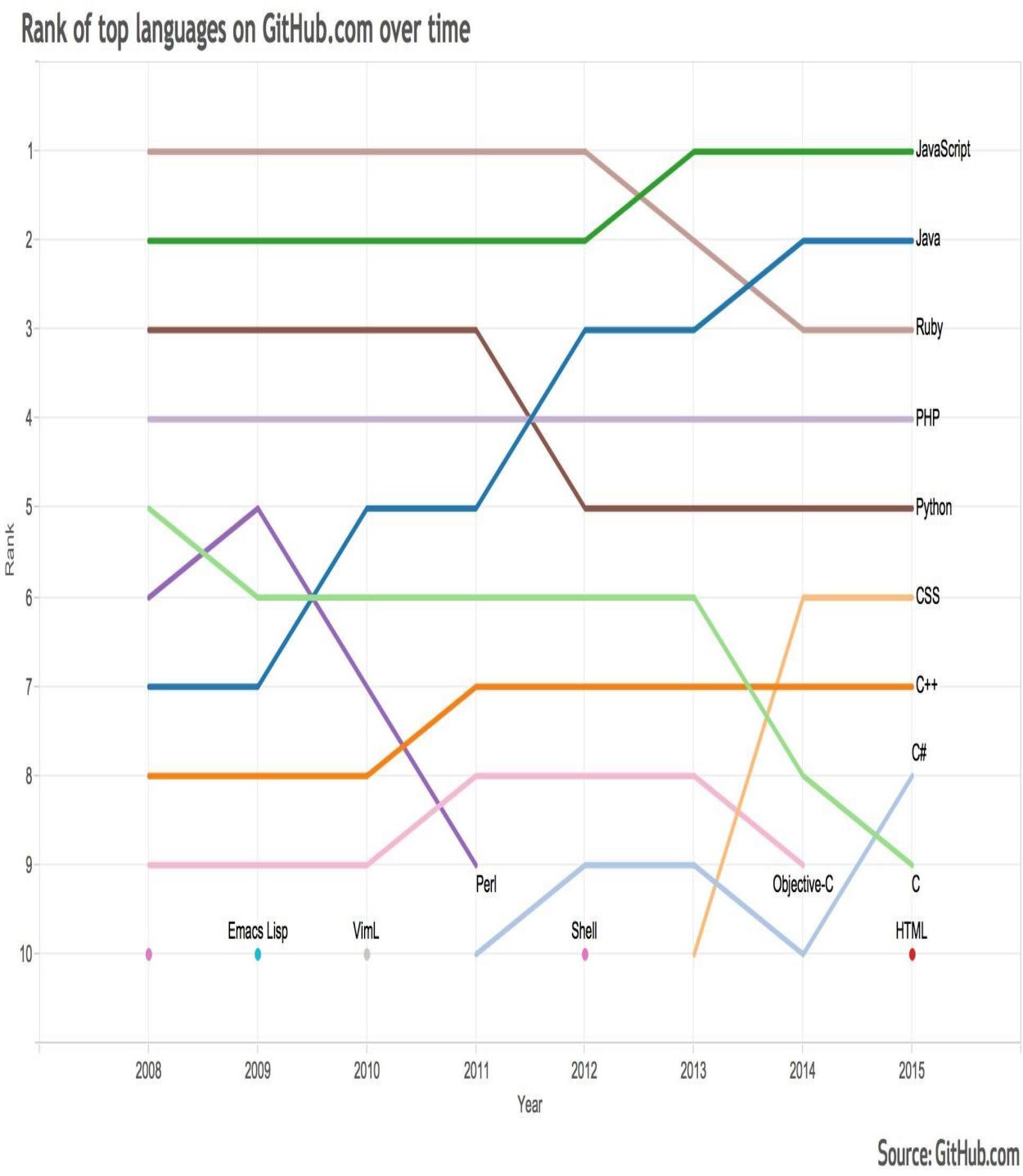
* Python is one of the most used languages in 2016, with major tech giants including *Google and Quora* being its users.
* It also has a huge open-source fanbase, currently ranking 5th in the github repository trends. Also, according to Google Trends, economically developed countries - US, China and South Korea - are its top-3 users, which further strengthens its position.

Python has a large number of web-frameworks, including scalability-proven and time-tested frameworks like Django, Flask and Pyramid.

*Django:*

Django is the most popular and extensive among all of the python frameworks. It is good for developing complex applications with many individual parts. Its major users include *Pinterest, Instagram, Mozilla, The Washington Times, Disqus, Bitbucketand Nextdoor*.

The popularity of the framework is also on a constant rise. Check the trends here:



[**Django CSRF Usage Statistics**](http://trends.builtwith.com/framework/Django-CSRF)

Surely, Django is a little slow in incorporating new trending feature like Job Queues(which can be fulfilled with 3rd party libraries like Celery), but the explicitness of the framework separates it from its competitors like RoR and Laravel, which rather prefers the automagic approach. Explicitness is enjoyed by many programmers who wants to know what is happening underneath the surface.

Summing up, I would say, things are looking bright for both Python(in general) and Django(in specific), and you will be in a safe place for the next 10–15 years (atleast), if you are investing you time in Python(and Django) presently, but don’t forget to take some time out to know what currently trending frameworks has to offer, who knows, it might suit your needs better.

## 1.4 Description

This project introduces students to basic web design using Python for the programming language HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). The feamework designing by using Django The project e does not require any prior knowledge of HTML or web design. Throughout the course students are introduced to planning and designing effective web pages; implementing web pages by writing HTML and CSS code; enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia; and producing a functional, multi-page website.

The course topical outline provides a summary of course topics that can be used as a guide when progressing through the course. Upon successful completion of this course, students will be able to:

Recognize and understand HTML web page elements

Know how to write HTML code

Understand and apply effective web design principles

Enhance web pages using text formatting, color, graphics, images, and multimedia

Incorporate forms into web pages

Understand and apply CSS to format web page elements

Plan, design, and publish a multi-page website

Students should have a general background in using a computer, managing files, and a basic knowledge of the Internet. Students should also be able to navigate to and within a website using a web browser such as Chrome, Firefox, Internet Explorer, or Safari. Students do not need to purchase any software for this course.

Upon successful completion of this course, students will have a good foundation in web design using HTML and CSS and will be prepared to study more advanced web design topics in the next level course, [Advanced Web Design](https://cty.jhu.edu/online/courses/computer_science/advanced_web_design_wd2.html).

Today, most websites are actually web applications, which offer **“dynamic”** web pages that can change their content in any number of ways. For instance, a webmail application allows the user to interact with it, displaying all sorts of different information, often while staying in a single webpage.

The idea behind creating a Python-driven web application is that you can use Python code to determine what content to show a user and what actions to take. The code is actually run by the web server that hosts your website, so your user doesn’t need to install anything to use your application; if the user has a browser and an Internet connection, then everything else will be run online.

**2. System Requirement Specification**

**2.1 Software requirement**

1. HTML – HTML refers to hyper text markup language.in this project we use for web page layout.HTML is document-layout and hyperlink - specification language used to design the layout of a document and to specify hyperlink.
2. CSS – CSS is a language that descri s the style of an HTML document. CSS describe be how HTML elements should be displayed. CSS stands for cascadind style sheet.
3. Bootstrap- Bootstrap ,the world”s most popular framework for building responsive mobile-first sites. We use bootstrap in our project ,also it is free and open source front end web frameworks, itconcerns itself with front end development only.
4. Javascript – Javascript is the programming language of HTML and the web. Javascript has many functions to develop a website.Javascript supporting object-oriented,imperative and declarative styles.
5. Django framework- Django is an opensource python web framework used for rapid development, pragmatic ,clean design and secure website. Django has its own naming system for all function. It is fast and simple.
6. Python - Python is a high level , general purpose programming language. Python has a design philosophy that emphasize code reliability. We use python because it can be used to create web application.

**2.2 Hardware requirement**

1. A machine containing an operating system and browser.

**3. Data Analysis**

**3.1 Data Dictionary**

1. Subscriber Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Column** | **Data Type** | **Primary Key** | **Nullable** |
| Subscriber | Id | Int | YES | NO |
| Subscriber | Name | Character | NO | NO |
| Subscriber | Email | Character | NO | NO |

2. Content Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Column** | **Data Type** | **Primary Key** | **Nullable** |
| Content | Id | Int | YES | NO |
| Content | content\_type | Character | NO | NO |
| Content | Branch | Character | NO | NO |
| Content | Semester | Character | NO | NO |
| Content | Subject | Character | NO | NO |
| Content | File | File | NO | NO |

3. User Model

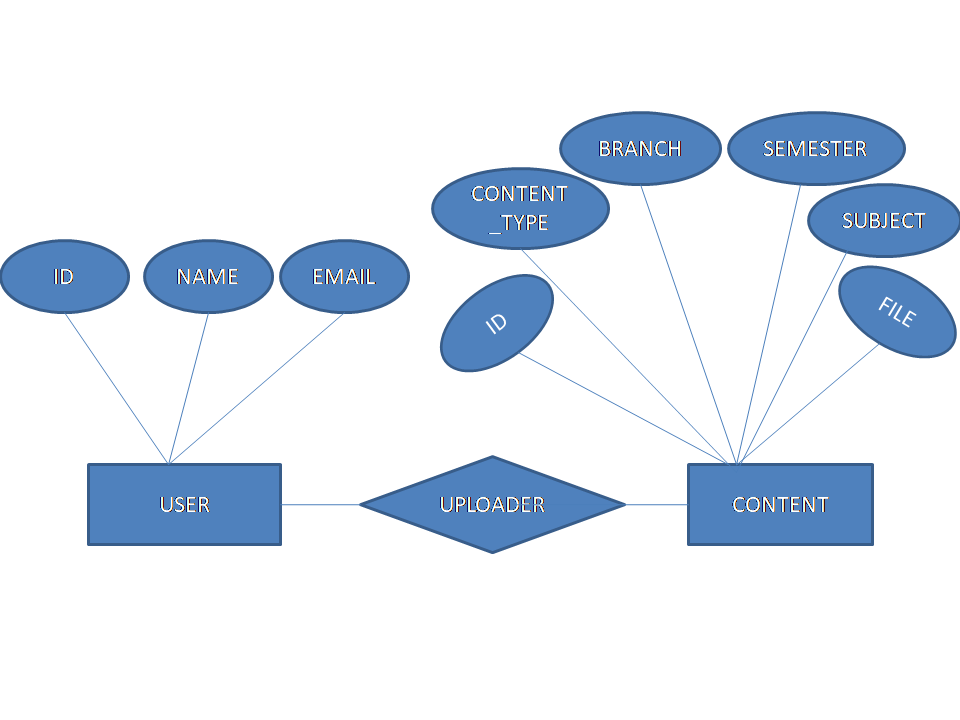
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Column** | **Data Type** | **Primary Key** | **Nullable** |
| User | Id | Int | YES | NO |
| User | Username | Character | NO | NO |
| User | Password | Character | NO | NO |

**3.2 ER diagram**

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity type.

An entity–relationship diagram for an MMORPG  using Chen's notation.

In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

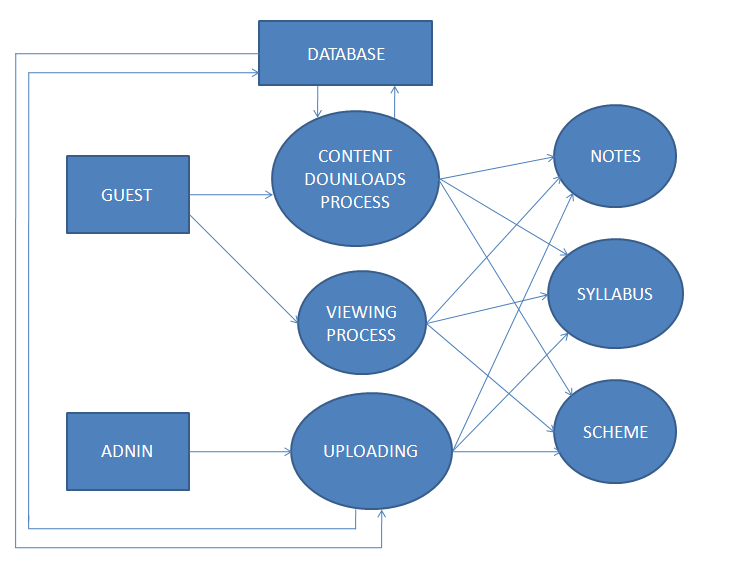
**fig. ER diagram**

**3.3 Data flow diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its  *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the  visualization  of  data processing(structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be

Stored . It does not show information about process timing or whether processes will operate in sequence or parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model



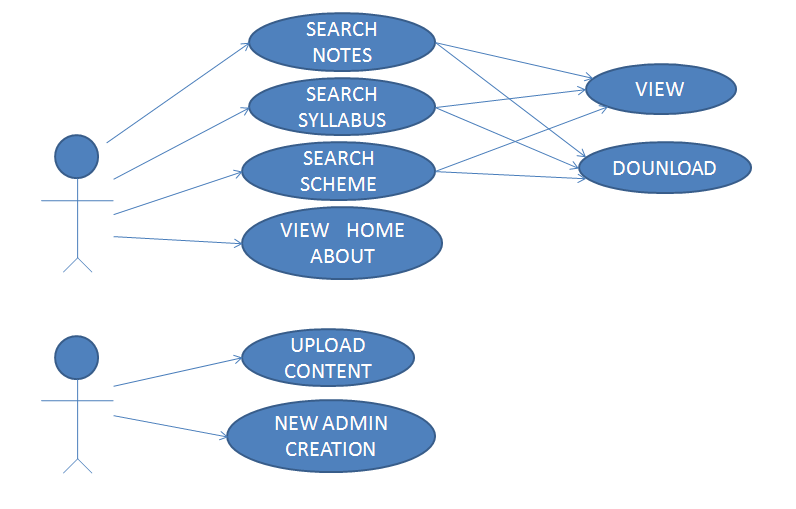
**Fig. Data Flow Diagram**

**3.4 Use Case diagram**

 A UML use case diagram is the primary form of system/software requirements for a new software program under developed. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple. It does not show the detail of the use cases:

* It only summarizes **some of the relationships** between use cases, actors, and systems.
* It does **not show the order** in which steps are performed to achieve the goals of each use case.

****  **fig. Use Case Diagram**

**4. Software Design**

**4.1 Development Model**

Agile mode; is used for the development of this web app. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

**4.2 Key Logic**

The main motive of our project is to provide subject notes to students which is very useful part of students’s life. Engineering Notes is a website which provides Useful  notes , scheme & Syllabus for students. Its mission is to offer Subject Notes much more in a unified and cohesive manner.

Everything is free and all the files are uploaded to our super fast servers so that they can be easily downloaded with high speed.

A User have to subscribe for accessing the files, all the files are free & universally accessible without any condition or restriction. Subscription will help user alot because we will provide mail for upcoming updates.

We will try our best to cover each and every content but still if you want some more information related to any of the posts then feel free to comment us.

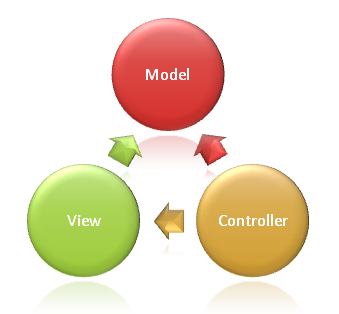
**4.3 Architecture**

The MVC pattern is a software architecture pattern that separates data presentation from the logic of handling user interactions(in other words, saves you stress:), it has been around as a concept for a while, and has invariably seen an exponential growth in use since its inception. It has also been described as one of the best ways to create client-server applications, all of the best frameworks for web are all built around the MVC concept.

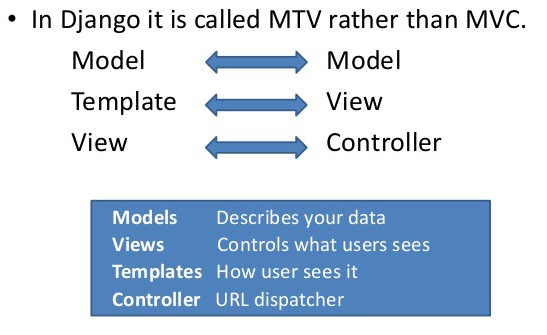
**Model**: This handles your data representation, it serves as an interface to the data stored in the database itself, and also allows you to interact with your data without having to get perturbed with all the complexities of the underlying database.

**View**: As the name implies, it represents what you see while on your browser for a web application or In the UI for a desktop application.

**Controller**: provides the logic to either handle presentation flow in the view or update the model’s data i.e it uses programmed logic to figure out what is pulled from the database through the model and passed to the view,also gets information from the user through the view and implements the given logic by either changing the view or updating the data via the model , To make it more simpler, see it as the engine room.

****

Django following the MVC pattern, it prefers to use it’s own logic in the implementation, the framework considers handling the Controller part of the MVC itself and letting most of the good stuff happen in the **Model-Template-View**, this is why Django is mostly reffered to as the **MTV**framework.

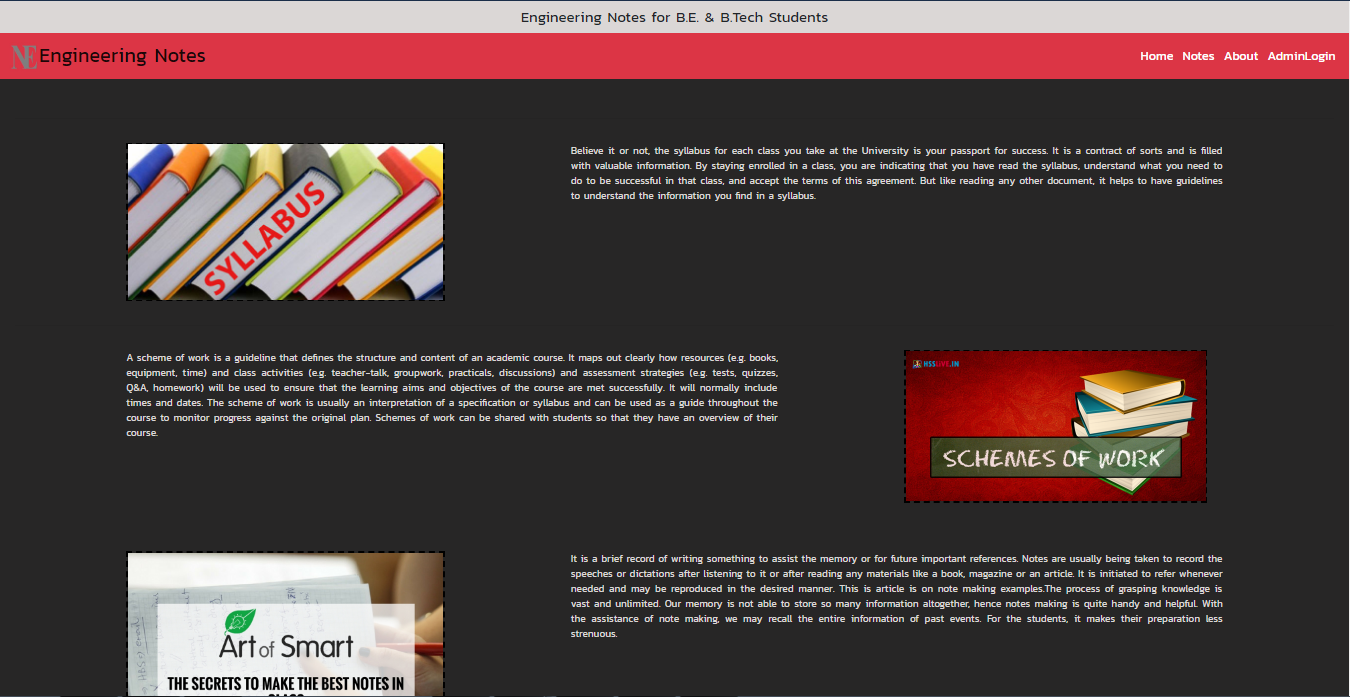
****

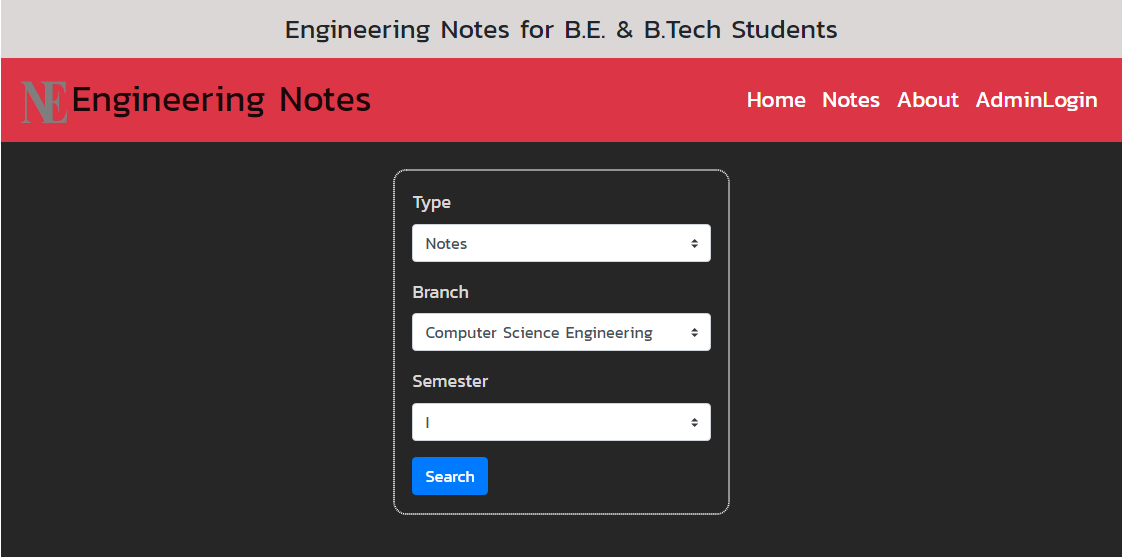
**Model**: Just like the Model explanation in the MVC pattern , this also takes the same position as the interface or relationship between the data and contains everything related to data access and validation.

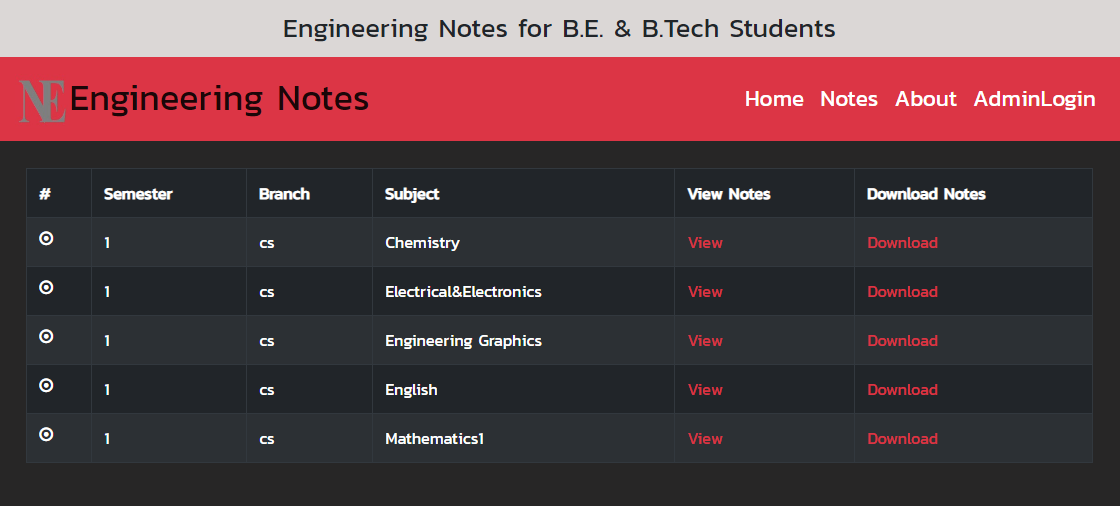
**Template**: This relates to the View in the MVC pattern as it is the presentation layer that handles the presentation logic in the framework and basically controls what should be displayed and how it should be displayed to the user.

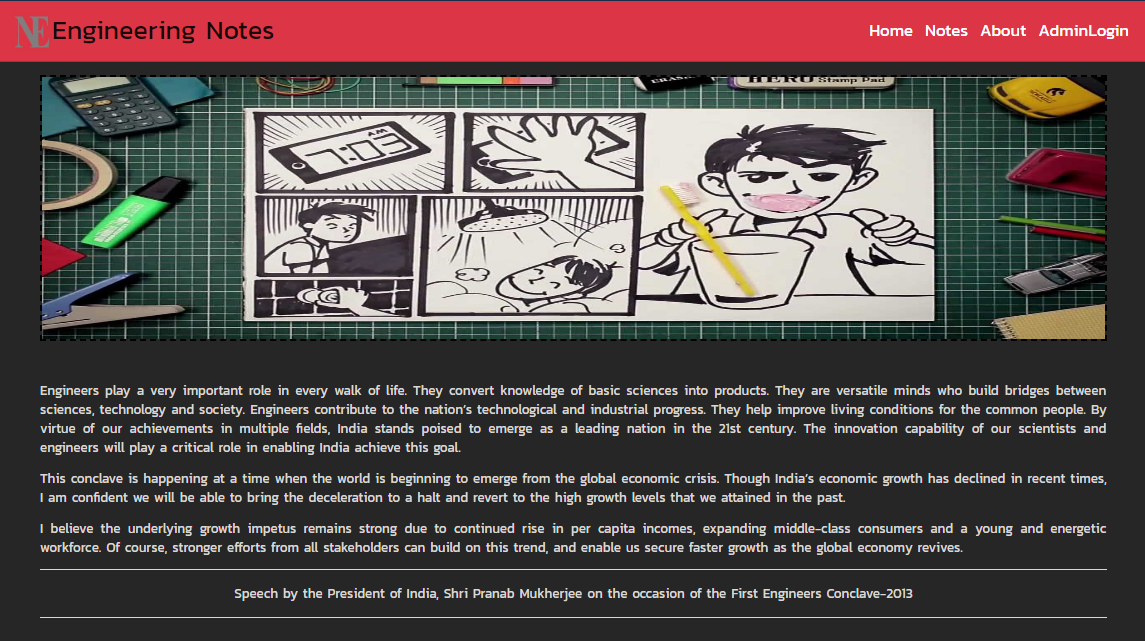
**View**: This part relates to the Controller in the MVC pattern and handles all the business logic that throws down back to the respective templates. It serves as the bridge between the model and the template.

**5. Snapshots**

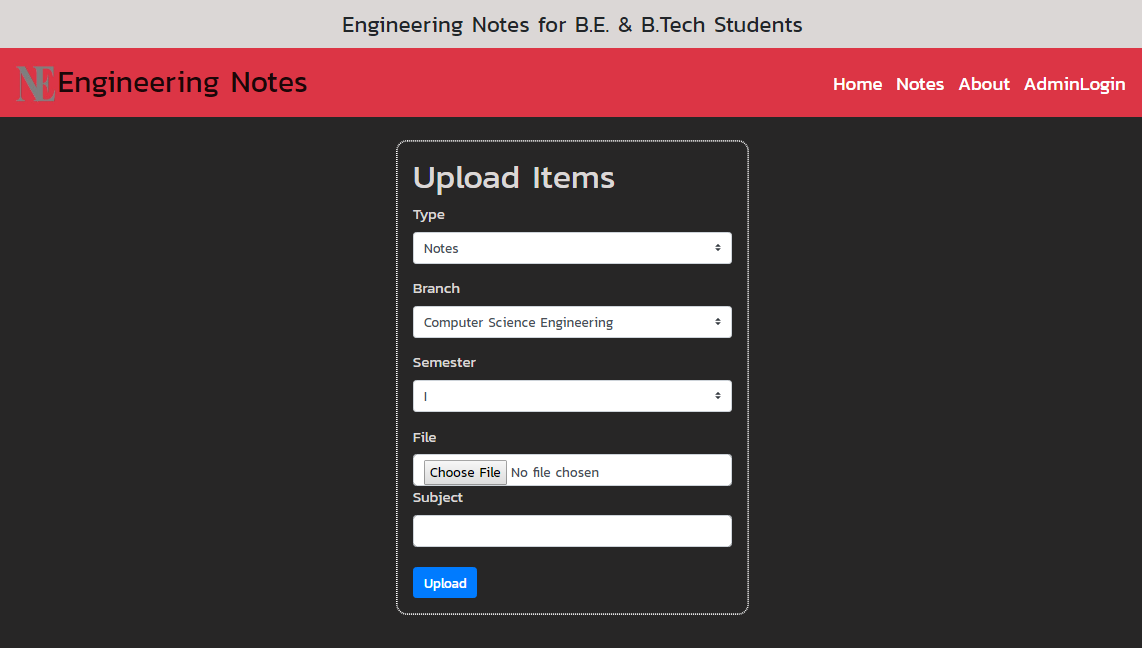
****

****

****

****

****

****

**6. Testing and Modification**

Test all links on your webpages are correctly woeking correctly and make sure there are no broken links.

\*scripting checkon the form are workingas expected.

\* Check default values are being populated.

\*Once submitted, the data in the forms is submitted to a live database or is linked to a working Email address.

\*Forms are optimally formatted for better readability.

\* Test HTML and CSS to ensure that search engines can crowl your site easily.

\* checking for syntax error.

\*usability testing has now become a vital part of any web based project.

\* Content should be legible with no speling or grammatical errors.

\* Images should correct.

\*Make sure queries sent to the database give expected results.

\*Test web server is handling all application requests without any service denial.

**DATABASE TESTING:**

\* Database is one critical component of your web application and stress

\* If any errors are shown while executing queries data must be laid to test it thoroughly. Testing activities will include-

\* Test Integrity is maintained while creating, updating or deleting data in database.

\* Check response time of queries and fine tune them if necessary.

\* Test data retrieved from your database is shown accurately in your web application

**Performance Testing**

## This will ensure your site works under all loads. Software Testing activities will include but not limited to -

* Website application response times at different connection speeds
* Load test your web application to determine its behavior under normal and peak loads
* Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.
* Test if a crash occurs due to peak load, how does the site recover from such an event
* Make sure optimization techniques like gzip compression, browser and server side cache enabled to reduce load times

**7. ADVANTAGES**

.

* As website name indicates ,all notes of engineering available at this platform
* Simple and secure and also reliable .
* The most important Easy to find and student can download easily.
* This application can run on the device own web browser through a simple URL thing that it is free.
* Quick and easy to upload.
* Reach anyone, anywhere in the world.
* you can also study without downloading odf files.
* This application run on mobile.
* And many more……

**8. Limitation and Future Enhancements :**

Following are the limitation of web applications :-

**Internet Reliance** – The one flaw with the internet is it is not everywhere yet, especially in many developing countries. If your internet goes down or you happen to be in an area that has not been connected yet you will not be able to access your web app

**Security** – There is no denying that your data is less secure when it’s in the cloud, especially when users from all over the world are accessing the same server hosted by a third party. Although there are ways to reduce your risk, email encryption and SSL enforcement for secure HTTPS access are just two examples.

**Reduced speed**– A web app will probably be slower than an application hosted on your company’s server. You need to decide if a slight reduction in speed is worth the worldwide access.

**Browser Support** – Unfortunately, we do not all use one version of a browser because we are given a choice. This means you will have to make sure your web app is supported across various browsers and for various screen sizes.

* Small screen size – This makes it difficult or impossible to see text and graphics dependent on the standard size of a desktop computer screen.
* Lack of windows – On a desktop computer, the ability to open more than one window at a time allows for multi-tasking and for easy reverts to a previous page. Historically on mobile web, only one page can be displayed at a time, and pages can only be viewed in the sequence they were originally accessed
* **Future Enhancement** :

**MARKUP:** Start out with semantic and well-structured HTML for flexibility and interoperability.

**STYLING:** Progressively enhance the look-and-feel of a design by adding support for browsers with greater features.

**BEHAVIOR:** Enhance the site with rich, interactive features on web browsers with JavaScript.

**9. CONCLUSION:**

The main Goal of our website is that every student easily interact with website and notes are easily available with just one click . which is very necessary for student that they know about syllabus and courses.

\* It is used to consume time

\* Easy to access

There are many reason building a website require that we can obtain the right tools and resources.

Standardization is one factor. When we use the right tools , we are able to view our webpages

No matter what browser they use. Our pages would be standard and acceptable to every visitor.

Our pages would load faster and we would also be able to maintain the pages we have created with ease.our pages become attractive if we use the right tools,

**10. REFERENCES**

* Roy Greenfeld, Daniel; Roy Greenfeld, Audrey (2015), [*Two Scoops of Django: Best Practices for Django 1.8*](http://twoscoopspress.com/products/two-scoops-of-django-1-8) (3rd ed.), Two Scoops Press, p. 531, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*0981467342*](https://en.wikipedia.org/wiki/Special:BookSources/0981467342)
* Jaiswal, Sanjeev; Kumar, Ratan (22 June 2015), Learning Django Web Development (1st ed.), Packt, p. 405, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*1783984406*](https://en.wikipedia.org/wiki/Special:BookSources/1783984406)
* Ravindrun, Arun (31 March 2015), Django Design Patterns and Best Practices (1st ed.), Packt, p. 180, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*1783986646*](https://en.wikipedia.org/wiki/Special:BookSources/1783986646)
* Osborn, Tracy (May 2015), Hello Web App (1st ed.), Tracy Osborn, p. 142, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*0986365912*](https://en.wikipedia.org/wiki/Special:BookSources/0986365912)
* Bendoraitis, Aidas (October 2014), Web Development with Django Cookbook (1st ed.), Packt, p. 294, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*178328689X*](https://en.wikipedia.org/wiki/Special:BookSources/178328689X)
* Baumgartner, Peter; Malet, Yann (2015), [*High Performance Django*](http://highperformancedjango.com/) (1st ed.), Lincoln Loop, p. 184, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*1508748128*](https://en.wikipedia.org/wiki/Special:BookSources/1508748128)
* Elman, Julia; Lavin, Mark (2014), [*Lightweight Django*](http://shop.oreilly.com/product/0636920032502.do) (1st ed.), O'Reilly Media, p. 246, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*149194594X*](https://en.wikipedia.org/wiki/Special:BookSources/149194594X)
* Percival, Harry (2014), [*Test-Driven Development with Python*](http://chimera.labs.oreilly.com/books/1234000000754) (1st ed.), O'Reilly Media, p. 480, [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*1449364829*](https://en.wikipedia.org/wiki/Special:BookSources/1449364829)
* Teachers guidance.
* Internet.