An Advanced Learner Project Report

ON

“Online Software Modelling for Continuous

Assessment Process”

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN

INFORMATION TECHNOLOGY

ENGINEERING

BY

Harish Natarajan (SE). UID: 118IT1400A

Dereck Jos Chakola(SE). UID: 118IT1153A

Akshar Vaskar(SE). UID: 118IT1001A

**UNDER THE GUIDANCE OF**

**PROF. Manivanan**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MGM COLLEGE OF ENGINEERING**

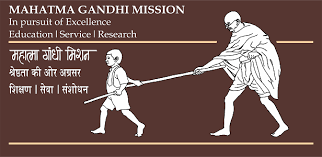
**KAMOTE, NAVI MUMBAI - 410209**

**2019-2020**

MGM COLLEGE of ENGINEERING

Department of Information Technology

KAMOTE, NAVI MUMBAI 410209



CERTIFICATE

This is certify that the project entitled

“ONLINE SOFTWARE MODELLING FOR CONTINUOUS ACCESSMENT PROCESS"

Submitted by

Harish Natarajan (SEIT) UID: 118IT1400A

Dereck Jos Chakola (SEIT) UID: 118IT1153A

Akshar Vaskar (SEIT) UID: 118IT1001A

Is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering ( Information Technology Engineering) at MGM College of Engineering, Navi Mumbai under the University of Mumbai. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. Manivannan) (Prof. K.V.Raman)

(Project Guide) (HOD, IT Department)

Acknowledgements

We are profoundly grateful to Prof.Manivanan for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards Dr. S.K.Narayan Khedkar, Principal, MGM College, Prof.K.V.Raman, Head of Department of Information Technology Engineering and Prof.Manivanan , Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Harish Natarajan (SE)

Dereck Jos Chakola(SE)

Akshar Vaskar (SE)

ABSTRACT

Currently the details of paper accessed is being entered manually by the faculty and generating report of this takes time. So here we are proposing a online system which could be used to enter the details and the reports could be generated automatically from the data entered.

There are three reports which would be generated 1) department wise report 2) faculty wise report 3) subject wise report. These reports can be easily downloaded

Contents

|  |  |  |
| --- | --- | --- |
| No | Topic | Page |
| 1 | Software Requirement and specification   * 1. Product Perspective ………………………..      1. Specific Requirement………………... | 7  7  7 |
| 2 | Project Plan  2.1 Designing home page/form…………………  2.2 Design link for  Registration and login page…………………  2.3 Log-in users to display summary…………...  2.4 Log-out design……………………………... | 8  9  11  12  13 |
| 3 | Design Strategy  3.1 Approach……………………………………  3.2 Django………………………………………  3.3 Model View Template (MVT)……………...  3.4 Running the application……………………. | 14  14  15  15  17 |
| 4 | Screenshots of project  Fig (a) Home page………………………………..  Fig (b) Link page…………………………………  Fig (c) Log in page……………………………….  Fig (d) Register page……………………………..  Fig (e) Welcome page……………………………  Fig (f) Summary page………………………….... | 18  18  19  19  20  20  21 |
| 5 | Conclusion and future scope  5.1 Conclusion…………………………………  5.2 Future scope………………………………. | 22  22  22 |
|  | References | 23 |

List of figures

|  |  |  |
| --- | --- | --- |
| No | Figure | Page |
| 1 | Form validation code in python | 10 |
| 2 | Link page HTML code | 11 |
| 3 | Python code to generate summary | 12 |
| 4 | Flow chart of actual working of model | 13 |
| 5 | Model Template View graph | 16 |
| 6 | Running Server using Command Prompt | 17 |

Chapter 1

Software Requirements Specification

1.1 **Product Perspective**

The purpose of this project work is to provide an efficient and easier way of storing data and generating a summary of the papers accessed. In the current system data is stored in excel sheets and manually the required tables are generated. A need for much simpler and easier way without much effort has aroused. So we aim at developing a web app which will prove to be more fruitful for the same.

.

1.1.1 **Specific Requirements**

The whole project is developed using Django Framework which uses python programming language for back- end and HTML5 and CSS3 for front-end. So the overall software requirements of the project are:-

* Django version 2.x
* Microsoft Visual Studio Code
* Windows XP or another Operating system

As an end product the web app developed by us will be implemented in mobile or a web browser. In that scenario the whole internal working will be abstracted from the users of the product as it is of no importance to them.

Chapter 2

Project Planning

The overall aspect of the project work has been to develop a proper and simple management of the data of assessed papers using a web application for the same

A detailed study of database and internet programming was conducted as an initial stage

The study of django, CSS and HTML5 have also helped in our development work

The major plan that will be executed for the purpose of designing are

* Designing home page/ form with validation
* Design a link for registration and login page
* logged in users are displayed the summary
* Logout to home page

2.1 **Designing home page/form**

The form is designed to take the user inputs such as name, semester, semester type (CBCGS or CBSGS), subject, etc.

While filling the form the user may input empty values which will be interpreted as NULL so at most care is to be taken to deal with nonoccurrence of NULL values that is a proper and efficient validation system is to be developed

The first step is to check which data is becoming frequently null. Then these values are set such that they cannot be null. Once this stage is completed, we have to check whether all the data entered are in correct format or not for example if the form is expecting integer but string values can be given as input then it creates an error.

If the above type of error occurs then we must construct a validation system that will also check for the data type of the field and only allow the required data type to be entered thus preventing from occurrence of invalid information

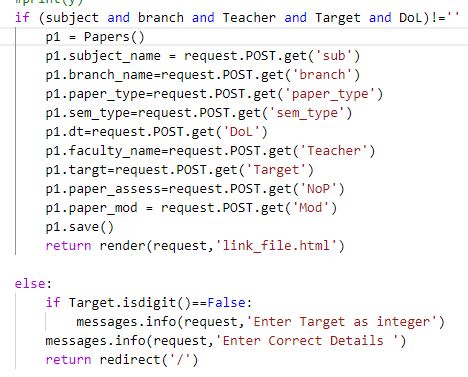


Fig 1. Form Validation code in python

2.2 Design Link for Registration and Login Page

Once the form is validated, the link page is opened. Here we get a login and register link along with a message that the form is submitted successfully. If the user is a valid user and he/she clicks on login link they get access to login page. After logging in the summary page is generated for them. If he/she is not a valid user then they can use the register link to register in the database. After successful registration login page opens. If the user is not a valid user and he/she clicks the login link then the registration page is opened by default and the user can register themselves.

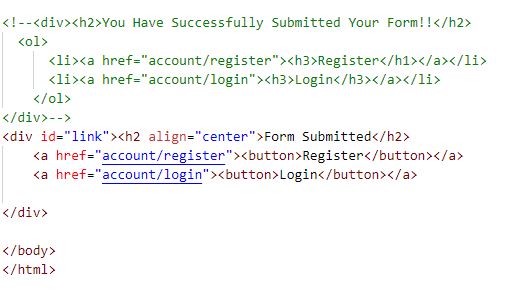


Fig 2. Link Page HTML code

2.3 Log-in users to view summary page

Once logged in the user can view the summary page. The summary is of three tables:-

* Faculty wise summary
* Department wise summary
* College wise summary

The admin can view any of these tables as per his/her wish. The faculty will get to see the faculty wise summary page. The admin can then check the number of papers assessed easily using the summary page. This summary can then be downloaded and sent to the University for their Reference. Since the tedious job of generating the report every time manually is avoided the work of admin is much more simplified.



Fig 3. Python code to generate summary

2.4 Logout design

Once the user is logged in and the summary is seen the user can log out of the application. Once the user clicks the logout button he/she is redirected to the home page again.

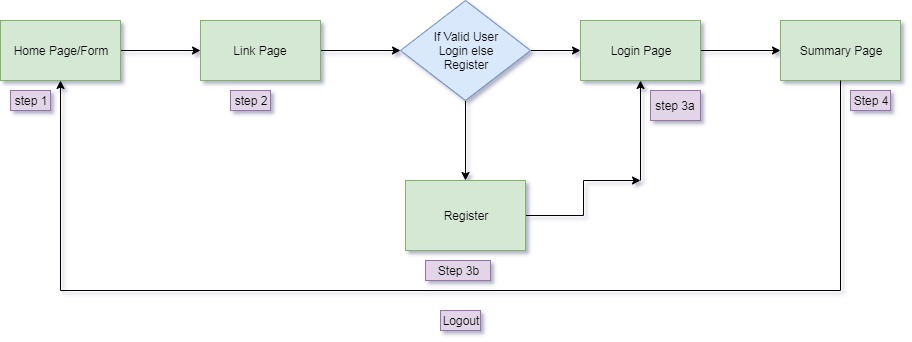


Fig 4. Flowchart of actual working of the model

Chapter 3

Design Strategy

The main design procedure in this project work of ours has been carried out with the model being done using Django which is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django is an open sourced framework. Development started in the year 2003, and since then different contributors have written several model for it. Django is used by companies like NASA, Instagram, Spotify, You Tube, etc. Before the entire working of the project took its full shape we first learnt the whole working of the database system, web programming and then we created this sample project work to check our knowledge. We have used the Django framework as it provides robust security features and it is easy to extend and scale.

3.1 Approach

First, the Django framework was needed to be studied. After that, the model was developed as a slow and gradual process. The database system was studied and it was implemented in Django.

3.2 Django

Django is an open sourced Python framework which was developed in 2003.Its primary use is to build complex web applications in a simpler and easier way.

It runs on both Unix and Windows. Other similar frameworks are spring, flask, lavarel.

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

3.3 Model View Template (MVT)

The MVT (Model View Template) is a software design pattern. It is a collection of three important component Model, View and Template.

Model:- The Model helps to handle the database. It is a data access layer which handles the data.

Template:- The Template is a presentation layer which handles User Interface (UI) part completely

View:- The View is used to execute the business logic and interact with the model to carry data and render the template.

Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the frame itself. There is no separate controller and the complete application is based on Model View and Template. That’s why it is called MVT application.

See the following graph that shows MVT control flow



Fig 5.Model Template View Graph

Here a user requests for a resource to the Django. Django works as the controller and check to the available resources in the URL. If URL maps, a view is called that interacts with model and template, it renders a template. Django responds back to the user and sends a template as response.

3.4 Running the Application

The method of running the application is much simpler. First we have to run the wsgi (Web Server Gateway Interface) light-weight server provided at the time of installing Django. The command to run the server is “python manage.py runserver”.If there are any errors in programming the command will output all the error in the logic in the cmd. After starting the server open browser and go to the URL <http://127.0.0.1:8000/> . Then you can check if the application works properly. If the template is not linked properly, it will display the error in the browser screen.

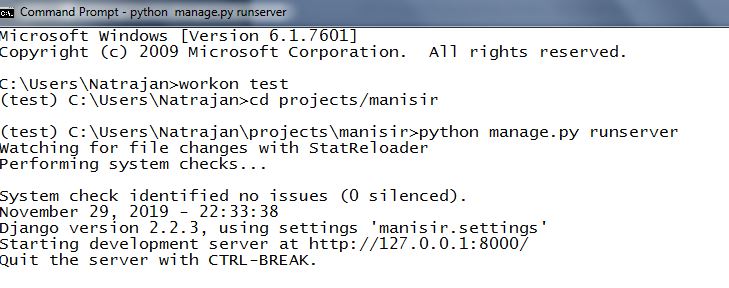


Fig 6. Running the server using Command prompt

Chapter 4

Screenshots of Project

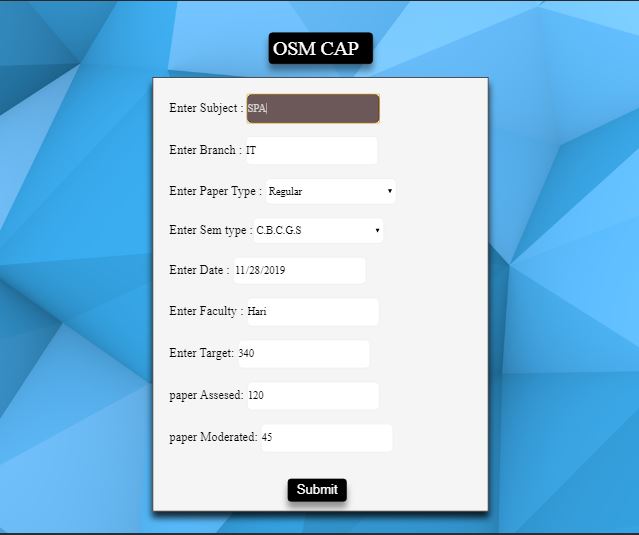


Fig (a). Home page

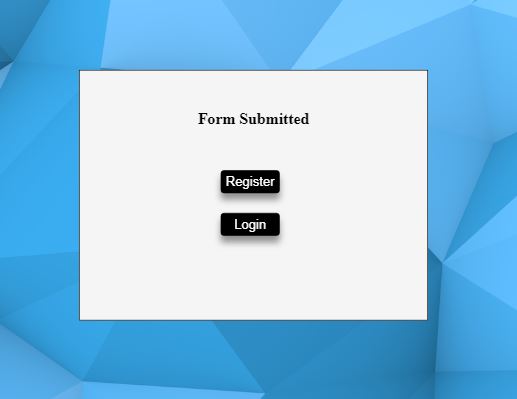


Fig (b). Link Page

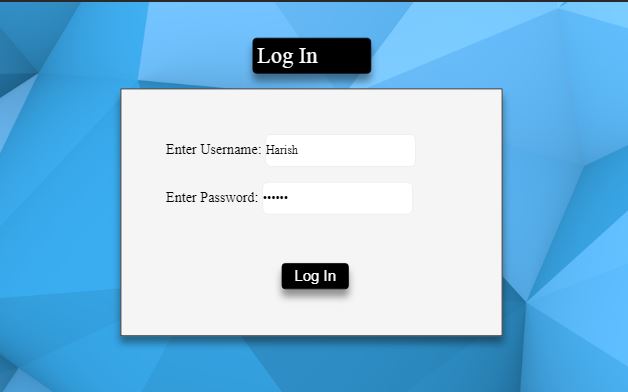


Fig (c). Log in page

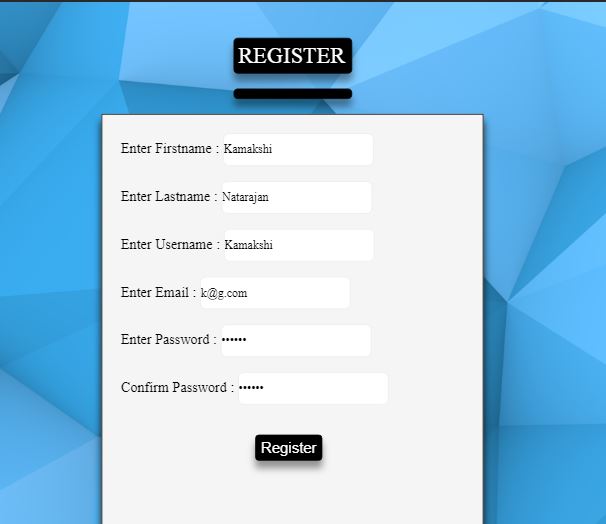


Fig (d).Register page

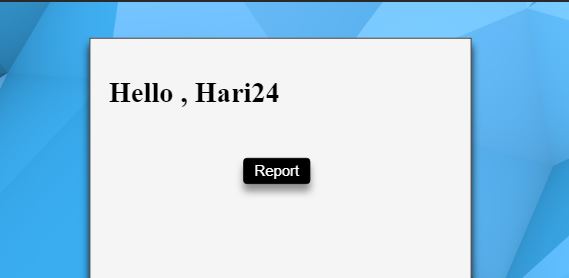


Fig (e). Welcome Page

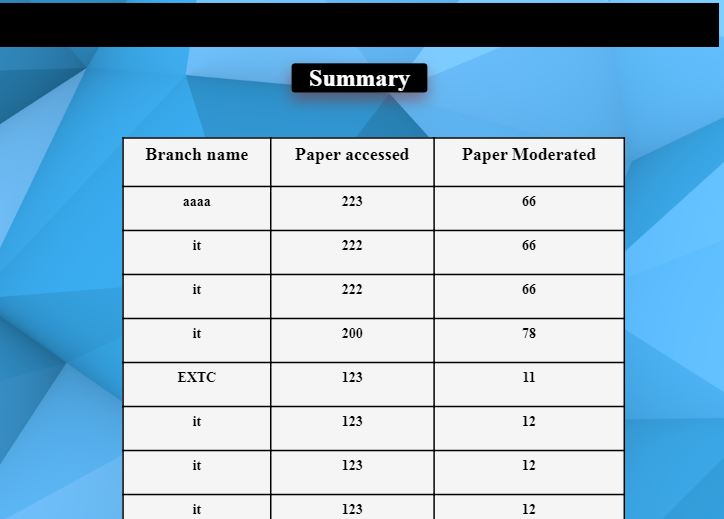


Fig (f). Summary Page

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The time and efforts of manually entering the data and generating report in excel sheet has been reduced as a much simpler and easier solution of filling a web form has been designed which can be used to generate these reports (faculty wise report, department wise report, college wise report) automatically. The generated reports can be downloaded and sent to the concerned authorities for their ready reference.

5.2 Future Scope

The current model runs only on web browser. Every time the user is required to go to the website and then fill all the details. So in future, the web version can be converted to a mobile app, doing so will simplify the model much more and it will also make it User-Friendly.

References

1. <http://nxgcloud.com/index.php/2018/04/04/django-read-data-database-show-html-page/>
2. <https://simpleisbetterthancomplex.com/tutorial/2018/11/28/advanced-form-rendering-with-django-crispy-forms.html>
3. <https://realpython.com/tutorials/django/>
4. <https://youtu.be/SlyxjRJ8VNY>
5. <https://simpleisbetterthancomplex.com/tutorial/2016/12/06/how-to-create-group-by-queries.html>
6. <https://www.learningaboutelectronics.com/Articles/How-to-insert-data-into-a-database-from-an-HTML-form-in-Django.php>
7. <http://www.learningaboutelectronics.com/Articles/How-to-print-out-all-rows-of-a-database-table-with-Python-in-Django.php>
8. <https://docs.djangoproject.com>
9. <https://www.djangobooks.com>
10. <https://www.tutorialspoint.com/css/index.htm>
11. <https://www.javatpoint.com/django-mvt>
12. <https://en.wikipedia.org/wiki/Django_(web_framework)>