**International Institute of Professional Studies**

**Devi Ahilya Vishwa Vidhyalaya**

**Indore, M.P.**

**Project Report**

**On**

**ONLINE FEEDBACK SYSTEM**

**This Project is submitted for VIII-semester**

for degree of

**Bachelor of Computer Application(HONS.) - 2015**

Guided By : Submitted By :

Mr. Jugendra Dongre Rohini Choudhary

IC-2K11-69

Swati Mahajan

IC-2K11-87

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “Online Feedback System” submitted to Devi Ahilya University in partial fulfilment of the requirement for the award of the degree of BACHELOR OF COMPUTER APPLICATION (BCA - Hons), is an original work carried out by Ms. Rohini Chaudhary(IC-2k11-69) and Ms. Swati Mahajan(IC-2k11-87) under my guidance.

The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other UNIVERSITY / Institute for the fulfillment of the requirements of any course of study.

Internal Examiner : External Examiner :

Signature: Signature:

Date: Date:

**RECOMMENDATION**

The Project wok entitled “**Online feedback System**” submitted by Rohini Chaudhary and Swati Mahajan is satisfactory account of the bonafide work under my supervision and is recommended towards the end of their I year of M.C.A. 2015.

**Guided By :**

**Mr. Jugendra Dongre**

**ACKNOWLEDGEMENT**

We acknowledge our sincere thanks to those who have contributed significantly to this project. It is a pleasure to extend deep gratitude to our internal guide Mr. Jugendra Dongre, IIPS, for his valuable guidance and support and to continuously prompt us for the progress of the project. We thank him for his valuable suggestions towards our project, which helped us in making this project more efficient and user friendly.

We thank and acknowledge each and every ones efforts that helped us in some or the other way for small and significant things.

**ABSTRACT**

The project aims at developing a completely Integrated Application designed to cater to every administrative requirement in the proper functioning of Feedback System. It will be a tool for managing resources, gathering essential data, conversion of this data into information, and making the information available to right person at right time for informed decision making. This includes receiving feedback of students, analysis of feedback and operating multiple entities without compromising on the integration of the application. Thus the project aims at improved service and consistent reviewing of status of the feedback at any point of time.

**TABLE OF CONTENTS**

**1. INTRODUCTION 8**

* 1. The Client Organization 9
  2. Problem definition 9
  3. Aim 9
  4. Objectives 9
  5. Project Goals 10
  6. Benefits 10
  7. Methodology 10

**2. CURRENT SYSTEM AND PROPOSED SYSTEM 13**

* 1. Current system 14
  2. Limitations of Current System 14
  3. Proposed system 14
  4. Objectives of Proposed System 15

**3. FEASIBILITY STUDY 16**

* 1. Feasibility Analysis 17
  2. Economical Feasibility 17
  3. Technical Feasibility 17
  4. Behavioral Feasibility 19

**4. ANALYSIS 20**

4.1 Questionnaire 21

4.2 Statistical Analysis 22

4.3 Interview 22

**5. PROJECT PLANNING 25**

* 1. Project Scope 26
  2. Document Plan 26
  3. Team Structure 27
  4. Project Deliverables 27
  5. Gantt Chart 27-28

**6. DESIGN 29**

6.1 Logical Design 30

61.1 Entity Definition 30

6.1.2 Attribute Definition 31

6.1.3 Relationship 35

6.1.4 E-R Diagram 36

6.1.5 Data Flow Diagram 37

6.1.6 Flow Diagram 37

6.1.7 Use case Diagram 38

6.2 Physical Design 39

6.2.2 User Interface 40

**7. IMPLEMENTATION 46**

**8. CODE SNIPPET 48**

**9. TESTING 54**

9.1 Black box testing 55

9.2 White box testing 55

9.3 Unit testing 55

9.4 Integration testing 56

9.5 System testing 56

**10. FUTURE PROSPECTIVE 57**

**12. BIBLIOGRAPHY 59**

**13. REFERENCES 59**

1. **INTRODUCTION**

**INTRODUCTION**



* 1. **The Client Organization**

The Online Feedback System is a college level feedback system which allows a student to provide their feedback online which reduces student’s headache of filling up the form in college on a paper. So this system allows him/her to provide feedback on their convenient time and place.

**1.2 Problem Definition**

The following points describe the problem domain at the client organization:-

* Non integration of all feedback's provided by students that belongs to different courses and having common subject or faculty.
* Paper work consumes a lot analysis time because to use automated analysis tools, first the paper work is needed to be converted into soft format.
* All the information in the system is maintained on the paper. This results in wastage of time and energy.
* Due to manual system, feedback action was not done timely.
* Current system requires faculty’s effort to collect all the students together and distribute them feedback form individually.

**1.3 Aim**

To design and develop a Feedback System for college, so as to make the feedback management more efficient, speed and easy by making analysis process as simpler as possible.

**1.4 Objective**

The objectives of the development are as follow:

* Integration of all feedback provided by students of different course.
* Computerization of present paper system which results in economic use of time and cost.
* Validation on number of program, semesters, batch id and section.
* Provides resume facility in case of any problem regarding resources like power off etc.
* To increase efficiency of current system.
* Provides the facility to administer to renew the form according to current by scenario by allowing him/her to update the questions, adding/removing batches etc.

**1.5 Project Goals**

* Reduce complexity of handling data.
* Implementing proper checks in every field.
* To provide an electronic version for feedback system.
* System will have a user friendly graphical interface and will be more cost effective as compared to non-electric version.
* Backup can be made on secondary memory like CD etc. .

**1.6 Benefits**

The various benefits that can be realized from this project are as follows:

* **Easy maintenance of information**

The proposed system maintains the details of the feedback provided by the student. Thus, it helps administer in efficient searching of information.

* **No dependency on a specific person to get required information**

Since the information will be available on server and can be accessed via proper authentication, dependency on a specific person will be removed.

* **Saves time and effort**

Since the feedback will be taken online, it will save the time and effort required in distributing form manually and recollect them and task of converting feedback into soft format for analysis via automated tools.

* **No involvement of pen and paper**

The proposed system do not make use of pen and paper since all the activities will be held via computer system. Number of semester students stays in hostel, each semester fees will be taken.

* **No restriction of place**

The proposed system allows students to provide their feedback online. Hence it avoids the restriction of place. Feedback can be provided from anywhere.

* 1. **Methodology**

The project plan is to carry the design and implementation of the project in a completely step-by-step manner. The entire project is divided in the following phases:-

**Phase 1 – Study and Analysis Phase**

* Questionnaire done to know the facilities and problem encountered by the student in providing feedback on paper.
* Statistical analysis done on the basis of questionnaire. Some drawbacks were encountered from this section.
* Questionnaire done to know the facilities and problem encountered by the faculty in taking feedback on paper and applying further action like analysis.
* Statistical analysis done on the basis of questionnaire. Some drawbacks were encountered from this section.

**Phase 2 - Design Phase**

In this phase the database design of the system are made. The database design is being carried in the following steps-

* Identified the entities and their relationships from the scenario.
* Designed ER model for the proposed system and relation among these entities were noted down in form of tables, having fields as attributes. (included in project report)
* The logical model of the system is designed, normalizing the relations.
* Data flow diagram of the whole system is constructed. For proper information flow, each level DFD is prepared. (included in project report)
* Flow chart of each process of DFD is constructed for better flow of data and its verification. (included in project report)
* Proposed the physical design of the system.
* Defining hardware and software requirements.

**Phase 3 - Coding Phase**

* The design of the system is implemented through actual code.
* Proper validation of data is used.
* Proper validation on important fields provided.
* User does not need to have the knowledge of the code, output is defined user friendly.

**Phase 4 – Testing and Implementation**

Testing is done on various test cases and data set. Testing is applied on:-

* Testing on each form by taking different user cases.
* Updation of record .

Testing is done by:-

Black box testing, White box testing and unit testing.

The above plan follows the Software Development Life Cycle (SDLC) model for the development of the proposed system. There are several ways for developing software. They consist of the following steps:

**Preliminary Investigation** - When a request to receive assistance from an information system is made, the first system activity, the preliminary investigation begins. It has three parts-

* **Request clarification** - All the request of the feedback system are met by clarifying each one with the faculties and authorities.
* **Feasibility analysis** - An important outcome, The system is technically feasible, college budget is kept in mind, so economically supportive, and implementation is not so difficult, so operationally feasible.
* **Request Approval** - Not all requests are desirable or feasible. So the faculty and authority decides which one are important and then we schedule them accordingly.

**Determination of system requirements** – We studied the Feedback process to answer the following questions:

* What is being done to take feedback and how is it being done?
* How frequently feedback request occur?
* How great is the volume of transactions?
* How well is task being performed?
* Does a problem occur in updation?
* How serious is the problem of searching?

For this questionnaire and survey are done. Conclusion are drawn from questionnaire.

**Design of a system** - It produces the details that state how the system will meet the requirements specified during systems analysis.

Design of report and output → Design of input data and form, and data to be stored → validation and verification code done→ testing.

**2. CURRENT SYSTEM AND PROPOSED SYSTEM**

**CURRENT AND PROPOSED SYSTEM**



**2.1 Current System**

The current system that is being used in the International Institute of professional studies is completely manual in nature. Feedback of various students is taken separately on papers. After examination, students are required to get collected in a class room then respective batch mentor(faculty) distributes the feedback form to each individual student. Students provide their feedback on that form by filling necessary details. Then Batch mentor collects that form and do the further action like analysis task.

* 1. **Limitations of Current System**

The current system of taking feedback has a large number of limitations. These limitations encourage the improper analysis of feedback. At the time of requirement gathering and analysis the following limitations are discovered-

* Since the system is manual there is scope of error due to carelessness and no manipulation.
* No backups of the records are maintained due to the tedious nature of the work. This might lead to loss of data in case of calamities like fires etc.
* Searching for particular records gets more and more difficult as the records get older.
* Analysis is done either manually, or provided feedback is converted into soft format to make use of automated analysis tools, both may result in error.
* The manual system is not consistent in nature.
* In case of requirement of multiple copies of a document, duplications have to be made by hand.
* Since all record is on paper, loss of data can take place due to wear and tear over a period of time.
* Important reports and comparison charts cannot be generated on time due to cumbersome nature of task.
* Students are required to remember their subject code and name of respective faculty to provide feedback about them.
* No recovery of data is possible by any means if it gets lost.
* To maintain all the records a large amount of money is required every year.
* Students are required to provide feedback at fixed time not according to their convenient time.
* Current system is not eco-friendly as it make use of paper.

**2.3 Proposed System**

The following points will give an overview of the proposed system :

* The proposed system is a completely automated system, which manages the entire operability of the feedback system.
* The system also generates appropriate analysis reports based on data collected.
* The proposed system helps the administer to retrieve the information as quickly as possible.
* The system provides backup plans to avoid data loss that is of high priority.
* It is Scalable so that in near future the system can be expanded and customized to the college’s needs.
* The system is reliable enough to perform in adverse conditions.
* The system provides a user-friendly interface with a realistic view.
* The system provides an error free environment.
* The system is highly maintainable .
* The system provides search facilities to find a specific entry from the database.
* The proposed system takes the feedback anonymously but make use of unique-id to disallow the unauthorized use.
* System generates feedbak ID for each feedback. It helps the student to resume its form in case of any inconvenience.
* It will keep in account the various subject of respective courses and also which faculty teaches which subject. Hence student are not required to remember all these things.
* Number of semester, batches and section in respective courses are also maintained.
* The system will not allow any student to fill duplicate information by providing them unique-id.

**2.4 Objectives of Proposed System**

The aim of the proposed system is to address the limitations of the current system. Following are the objectives of the proposed system:

* Reduce data duplication.
* Implement validation techniques and checks that will help reduce the margin of error in operations.
* Provides adequate data backup facilities in order to ensure system restart even after a calamity.
* Since the system is Django(python framework) based, searching and cataloging of data is not a problem.
* The system ensures consistency.
* Should arrive at and obtain a complete automation of feedback system.
* The system is a reusable and extensible model/code.
* The system is a foolproof system that simulates and replaces the present manual system.

**3. FEASIBILITY STUDY**

**FEASIBILITY STUDY**



**3.1 Feasibility Analysis**

Feasibility study examines how beneficial is the project economically, technically and non- technically. The online feedback system is a truly feasible project and satisfies all the three conditions described below.

**3.2 Economic Feasibility**

The economic analysis of the project counts for the cost effectiveness of the project.

The economic feasibility of online feedback system is measured under the following heads:

* The project fits in the proposed budget.
* The project satisfies the economic constraints.
* The existing hardware meets the budget proposed.

This project can be regarded as economically feasible because the college possesses the computer in their computer lab, so there is no need of bearing any cost on the resources needed for the development of the system. The only cost involved in the project is the deployment cost. The only resource required for proper and successful deployment is a virtual private server meeting software and hardware requirements of the system. These costs are quite minimal and easily available with the organization. In addition, there are no start-up costs for communications equipment installations, recruitment of new personnel, cost of disruption to the rest of the system.

**3.3 Technical Feasibility :**

The criteria for technical feasibility of online feedback system are:

* **Easy to use :** The students and administer were assumed to be normal ordinary people having knowledge of simple computer operation, so ease of use is a very essential factor if we want to get the job actually done quickly.
* **Security:** As unique id is required to students to provide feedback so security is maintained. Some functions are to be kept under control of some specific authority i.e administer so for hm/her password protected entry is used.
* **Binary Compatibility:** Our software makes it possible to extend an existing class or interface by adding new methods and properties and yet still retain binary compatibility.
* **Reliability:** As tables are used for storing the data, so reliability is maintained as exact data is retrieved. Data is regained from each table by using a unique key so the accurate record is accessed.
* **Maintainability :** Maintenance usually comprises of 2 key factors:-

**Updation :-** As the project is build with keeping in mind all the necessary functions so no essential updation is required. Still if updation are required then administer have provided the capability to updation .

**Maintanance :-** Systems must be updated with antivirus so the proper functioning of operating system and browsers can be maintained.

* **Portability:** As the size of project code is enormous. Hence it is maintained on version control system and solves the portability issues. If required then data can also be moved using secondary memory devices. So it is portable to a large extent.
* **Extensibility:** The project is easily extensible as Django framework support further editing, such as adding new functions, new forms or editing old forms. Since it is maintained via version control system, updation is easy and lossless.
* **Reusability:** The extensibility and portability of this project would make it reusable software.
* **Serviceability:** This project will try to provide maximum possible service to the user.

In our project the technical feasibility is considered up to a great extend. The software is build using **Django framework**, which is freely available. Thus the problem of non- availability of software is eradicates. The backend of the system is MySQL, which is freeware database application. Proposed system can be expanded in future if required. Proposed system has the capacity to hold the data of the firm.

It also provides the data security by password protecting.

**Technology**

The front end of this project is designed using bootstrap and JavaScript.

Some major features of the above languages and tools are described below.

**Bootstrap**

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. **Bootstrap** is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It is the most popular project on GitHub and has been used by NASA and many others. It makes the web pages mobile responsive. Currently the IIPS site is also implemented on bootstrap.

**My SQL**

Many programming languages with language-specific APIs include libraries for accessing My SQL databases. The official My SQL Workbench is a free integrated environment developed by My SQL enables users to graphically administer My SQL databases and visually design database structures. My SQL Workbench replaces the previous package of software, My SQL GUI Tools. Similar to other third-party packages, but still considered the authoritative My SQL front-end; My SQL Workbench lets users manage the following:

Database design & modeling.

* SQL development – replacing My SQL Query Browser.
* Database administration – replacing My SQL Administrator.
* In this project All database created on the MY SQL .the information of examine will be stored on to the MY SQL tables and maintained as a dynamically according to new information.
* Student registration and login database will also be inserted into MY SQL database.

**Django**

**Django** is a free and open source web application framework, written in Python, which follows the model–view–controller (MVC) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a non-profit. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes re usability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. 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.

Some well-known sites that use Django include Pinterest, Instagram, Mozilla, The Washington Times, Disqus, the Public Broadcasting Service, and Bitbucket.

**Crispy-Form**

django-crispy-forms is a Django application that lets you easily build, customize and reuse forms using your favourite CSS framework, without writing template code and without having to take care of annoying details. You are currently looking at the documentation of the development release.

**3.4 Behavioural Feasibility :**

Behavioural feasibility of online feedback system is the measure that how effective the client uses the system. It is one of the major factors of feasibility analysis. The new and the proposed system- Online Feedback System is

* Easy to operate,
* Convenient in maintenance and
* Effective in its work.

Thus behavioral feasibility is very important factor to be considered for effective working of system.. The systems analyst must still consider the behavioral feasibility of the requested Online Fedback System. It is not dependent on the human resources available for the Online Feedback System and involves projecting the system operates and be used when deployed. The system is behaviorally feasible if it fulfills the following points :

* The Online Feedback System is easy to operate.
* In all the areas of application the expanded results are better than the earlier.
* Individual performance is not expected to deteriorate after implementation.
* Retrieval of information is easy, accurate and fast.

**4. ANALYSIS**

**ANALYSIS**



**Study and Analysis**

**4.1 Questionnaire**

Questionnaire done to know:-

* Problem encountered by the student in providing feedback.
* The facilities provided in his actual routine.
* Action taken or handled if he makes any complain.

**4.2 Statistical analysis**

Statistical analysis done on the basis of questionnaire to know:-

* Many drawbacks of present system came into light.
* Fields to be included or excluded are determined.
* Analysis time of feedback gathered.

**4.3 Interview**

Interview of faculty and authority taken, to know:-

* Present system functionalities were completely understood.
* Steps of action taken when a student provides feedback.
* Method of analysis was determined.
* Information regarding the project was taken.

**Questionnaire**

Q.1) Is the feedback process tedious? Yes  No

Q.2) Are you in favour of automation of feedback \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q.3)Which type of inconvenience occurs during feedback process \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q.4)Are you in support of eco-friendly nature of feedback System? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q.5)Are you frustrated from providing feedback using pen and paper ? Yes  No 

If yes then please give suggestions? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q.6)Will the online feedback system help you in providing feedback?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Statistical Results of Online Feedback System**

**Total forms:-40**

Q.1) Is the feedback process tedious?

Yes:- 22 (55%)

No:- 18 (45% )

Q.2) Are you in favour of automation of feedback?

Yes:- 32 (80%)

No:- 8 (20%)

Q.3) Which type of inconvenience occurs during feedback process?

Inconvenience:- 32 (80%)

Convenience :- 8 (20%)

Q.4 Are you in support of eco-friendly nature of feedback System?

Yes:- 26 (65%)

No:- 14 (35%)

Q.5) Are you frustrated from providing feedback on pen and paper ?

YES:- 16 (40%)

NO:- 24 (60% )

If yes then give suggesions?

Suggestions:- 30 (80%)

No suggestions:- 10(20%)

Q.6) Will the online feedback system help you in providing feedback?

Yes:- 8 (20%)

No:- 32 (80%)

**5. PROJECT PLANNING**

**PROJECT PLANNING**



The most critical phase of managing system projects is planning. Information systems have become increasingly important during the past decade. First, information is now recognized as a vital resource. Second, more and more financial resources are committed to information systems. Third, there is growing need for formal long-range planning with Information systems.

**5.1 Project Scope**

The scope of the project can be described in the following manner:

Designing a mechanism which can act as a platform for student to efficiently and easily provide feedback.

Designing a user friendly graphical user interface to enter programme, batch, semester and section information.

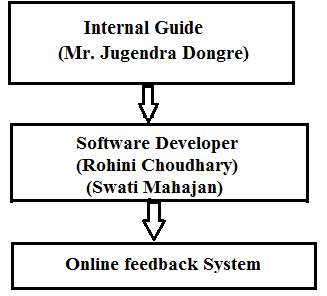
To develop a analysis on the basis of feedback provided by students.

**5.2 Development Plan**

The design and implementation of this project has been carried out in a completely step-by-step manner.

* Questionnaire
* Design Document
* Software Requirement Specification
* Coding
* Testing
* Project Report

**5.3** **Team Structure**



**5.4 Project Deliverables**

**Project Report**

Project report provides an introduction to current system and system to be built. It gives the brief knowledge of current system

**Project Documentation**

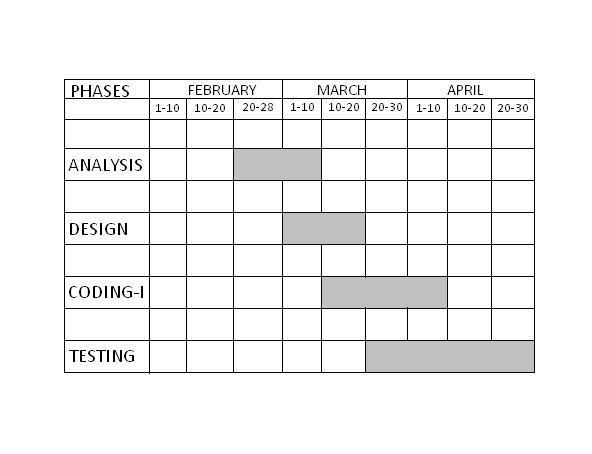
A complete documentation is given in the form of SRS. Software Requirement Specification (SRS) it provided an introduction of the current system and the system to be built. This overview includes the purpose of SRS and a brief knowledge of the current system. The overview also provided an introduction of the proposed system.

**Pre- Installation**

The installation of prior software’s like Microsoft Access. These software are essential for the smooth and proper running of project.

**5.5 Gantt chart**

Gantt chart is a time-phased bar chart display that lists tasks or activities along the left side and a corresponding bar for each task. The length of the bar represents the duration of the activity. This scheduling tool used to display the status of a project’s tasks. The Gantt chart shows each task’s duration as a horizontal line. The ends of the lines correspond to the task’s start and end dates.



**6. DESIGN**

**DESIGN**



**6.1 Logical Design**

Logical design describes the functions required of a system, those are:-

* Student authorization
* Student’s anonymous details
* Feedback about infrastructure
* Feedback about subjects and faculty
* Admin authentication
* CRUD facility to Admin
* Resume functionality of feedback form

The following are the objectives of Logical Design :

* Analyze functioning of feedback form, resume functionality, admin authentication and working of CRUD functionality.
* The ER Diagrams constructed by keeping in my general entries related to student and relation with other entities.
* Transformation of Conceptual Data Model into a Relational Model with data specifications.

**6.1.1 Entities Definitions**

Entities are the principal data object about which information is to be collected. Entities are either concrete or abstract, such as person, places, things, or events, which have relevance to the database.

The following is the **list of entities** that have been identified during analysis:-

* **Name:** Feedback\_student\_info  
  **Definition:** there can be large number of students

**Identifier:** fs\_id

* **Name:** Infrastructure\_support\_info **Definition:** contain information about infrastructure feedback

**Identifier:** id

* **Name:** Academic\_assessment\_info **Definition:**  contain information about academic feedback

**Identifier:** id

* **Name:** Course **Definition:** contains information about courses

**Identifier:** course\_id

* **Name:** Subject **Definition:** contains information about subjects with respect to courses

**Identifier:** subject\_id

* **Name:** Batch **Definition:** contains information about batches

**Identifier:** subject\_id

* **Name:** Time\_table **Definition:** contains information about which teacher taught which subject

**Identifier:** id

* **Name:** Faculty **Definition:** contains information about faculty

**Identifier:** faculty\_id

* **Name:** Section **Definition:** contains information about sections in respective batches.

**Identifier:** id

* **Name:** Question **Definition:** contains information about questions to be displayed.

**Identifier:** id

**6.1.2 Attribute Definitions**

The attributes that are identified as part of the entities are listed along with their description.

1. **Name of Entity:** Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Fs Id** | This is the unique id of the student. | Not null | Varchar2 |
| **2** | **Batch Id** | It shows a particular subject's batch id | Not null | Varchar2 |
| **3** | **Course** | Course in which student studies. | Not null | Varchar2 |
| **4** | **Semester** | Semester of the student. | Not null | Varchar2 |
| **5** | **Section** | Section of the Student. | Not null | Varchar2 |
| **6** | **Feedback Session** | Year of the Feedback . | Not null | Varchar2 |

**2. Name of Entity:** Infrastructure\_support\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Id** | This is the first name of student. | Not null | Varchar2 |
| **2** | **Books Availability** | Availability of books library. | Not null | Varchar2 |
| **3** | **Basic Requirements** | Basic requirement in a class. | Not null | Varchar2 |
| **4** | **Technological support** | Technological support provided by the college | Not null | Varchar2 |
| **5** | **Study Material** | Study material provided by the faculty. | Not null | Varchar2 |
| **6** | **Resource Availability** | Availability of all the required resources. | Not null | Varchar2 |
| **7** | **Cleanliness of class** | Cleanliness of the classroom. | Not null | Varchar2 |

1. **Name of Entity:** Academic\_assessment\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Id** | This is the Id of the student. | Not null | Varchar2 |
| **2** | **Subject Id** | It shows a particular subject's id | Not null | Varchar2 |
| **3** | **Faculty Id** | It shows id of the faculties. | Not null | Varchar2 |
| **4** | **Conceptual clarity** | Ability of Conceptual clarity by the facility. | Not null | Varchar2 |
| **5** | **Subject knowledge** | Subject Knowledge of Faculty. | Not null | Varchar2 |
| **6** | **Practical example** | Practical example given by faculty. | Not null | Varchar2 |
| **7** | **Handling capability** | Capability to handle class | Not null | Varchar2 |
| **8** | **Motivation** | Motivation provided by faculty. | Not null | Number |
| **9** | **Control ability** | Ability to control class. | Not null | Number |
| **10** | **Course Completion** | Ability to complete course on time. | Not null | Varchar2 |
| **11** | **Communication skill** | Communication skill of the faculty. | Not null | Varchar2 |
| **12** | **Regularity and punctuality** | Regularity of a faculty. | Not null | Varchar |
| **13** | **Outside Guidance** | Interaction outside the class. | Not null | Varchar2 |
| **14** | **Syllabus Industry Relevance** | Industrial knowledge of the syllabus. | Not null | Number |
| **15** | **Sufficiency of course** | It stores 12th % of the student. | Not null | Varchar2 |
| **16** | **Suggestions for subject** | Suggestions by the Student, if any | Not Null | Varchar2 |
| **17** | **Suggestions for subject** | Suggestions by the Student, if any | Not Null | Varchar2 |

4**. Name of Entity:** Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Course Id** | Course id of the student | Not null | Varchar2 |
| **2** | **Course Name** | Name of the course. | Not null | Varchar2 |
| **3** | **Stream** | Shows that course belongs to which stream like computer Science or management. | Not null | Varchar2 |

5**. Name of Entity:** Batch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Batch Id** | Batch Id of the student. | Not null | Varchar2 |
| **2** | **Session** | Year of the batch. | Not null | Varchar2 |
| **3** | **Course Id** | Course id of the student. | Not null | Varchar2 |

6**. Name of Entity:** Subject

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Subject Id** | This is the Id of the subject. | Not null | Varchar2 |
| **2** | **Name of Subject** | Name of the subject. | Not null | Varchar2 |
| **3** | **Semester** | Semester of the student. | Not null | Varchar2 |
| **4** | **Credits** | Credits of the subject. | Not null | Varchar2 |
| **5** | **Status** | Shows Subject is currently active or not. | Not null | Varchar2 |
| **6** | **Lab Viva** | It shows whether the subject has a lab viva. | Not null | Varchar2 |
| **6** | **Course Id** | Course id the student. | Not null | Varchar2 |

7**. Name of Entity:** Section\_information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Id** | This is the Id of the student. | Not null | Varchar2 |
| **2** | **Section** | Section of the student. | Not null | Varchar2 |
| **3** | **Batch Id** | It shows batch id of the student. | Not null | Varchar2 |

8**. Name of Entity:** Faculty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Faculty Id** | This is the Id of the faculty. | Not null | Varchar2 |
| **2** | **Name** | Name of the faculty. | Not null | Varchar2 |
| **3** | **Qualification** | Qualification of the faculty. | Not null | Varchar2 |
| **4** | **Date of birth** | Birth date of faculty. | Not null | Varchar2 |
| **5** | **Email** | Email id of the faculty. | Not null | Varchar2 |
| **6** | **Discipline** | Discipline of the faculty. | Not null | Varchar2 |
| **7** | **Gender** | Gender of the faculty. | Not null | Varchar2 |
| **8** | **Area of interest** | Area of interest of the faculty.. | Not null | Number |
| **9** | **Contact Number** | Contact number of the faculty. | Not null | Number |
| **10** | **Type** | Shows whether faculty is visiting faculty or permanent. | Not null | Varchar2 |
| **11** | **Status** | It shows date of birth of student. | Not null | Varchar2 |
| **12** | **Designation** | Designation of the faculty. | Not null | Varchar |
| **13** | **Responsibility** | Responsibility of the faculty. | Not null | Varchar2 |

9**. Name of Entity:** Time\_table

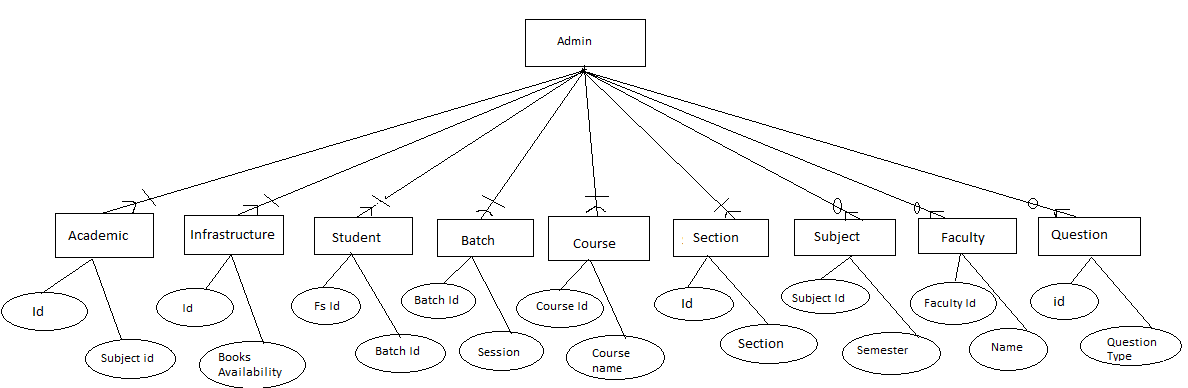
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **S\_no** | Unique id. | Not null | Varchar2 |
| **2** | **Year** | It shows a particular subject's id | Not null | Varchar2 |
| **3** | **Section** | It shows id of the faculties. | Not null | Varchar2 |
| **4** | **Day** | Teaching day of the subject. | Not null | Varchar2 |
| **5** | **Time From** | Starting time of the lecture. | Not null | Varchar2 |
| **6** | **Time to** | End time of the lecture. | Not null | Varchar2 |
| **7** | **Room Number** | Room alloted. | Not null | Varchar2 |
| **8** | **Batch Id** | Shows batch id. | Not null | Number |
| **9** | **Course Id** | Shows course id. | Not null | Number |
| **10** | **Faculty Id** | Shows faculty id. | Not null | Varchar2 |
| **11** | **Subject Id** | Shows subject's id. | Not null | Varchar2 |

10**. Name of Entity:** Question

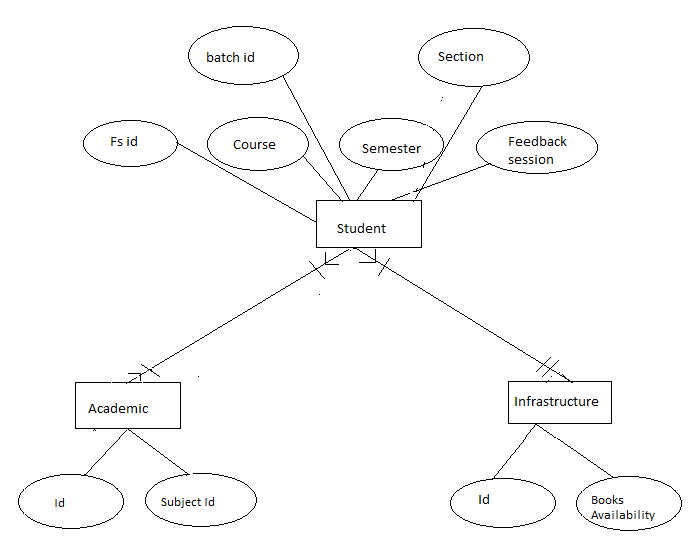
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Attribute**  **Name** | **Description** | **Constraint**  **name** | **Data type** |
| **1** | **Id** | This is the Id of the Question. | Not null | Varchar2 |
| **2** | **Question Text** | Text of the question. | Not null | Varchar2 |
| **3** | **Type** | Question type | Not null | Varchar2 |

**6.1.3 The entire E-R Diagram**

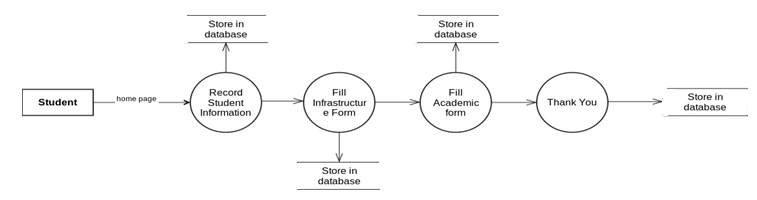
1. Admin

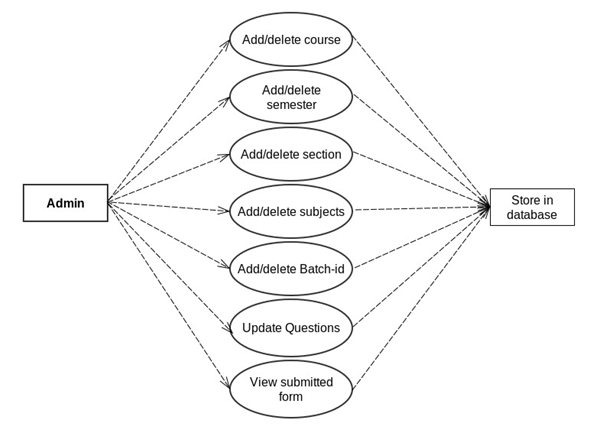


1. Student

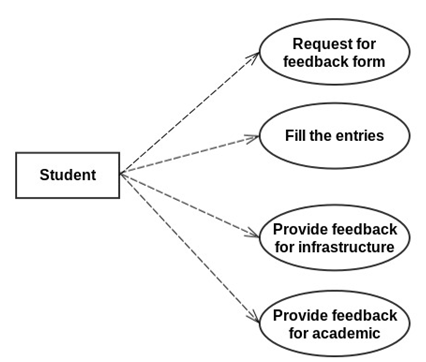
****

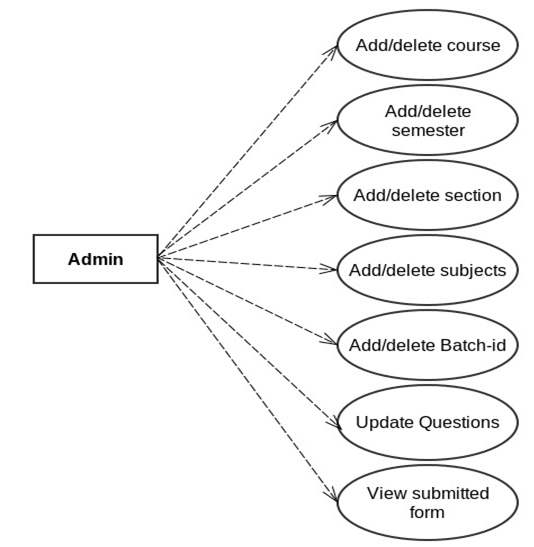
**6.1.5 Data Flow Diagram**





**6.1.7 Use case Diagrams**





**6.2 Physical Design**

* We have 10 tables at designed for storage.
* Normalized relations, to get database which can be stored on secondary memory.
* Tables are used for: - Entered, Retrieved, Deleted and Updated
* The point of response time:- Data security, Backup, Recovery, Retention and Integrity; are kept in mind.
* Backend files are also password protected.

**6.2.2 User Interface Design**

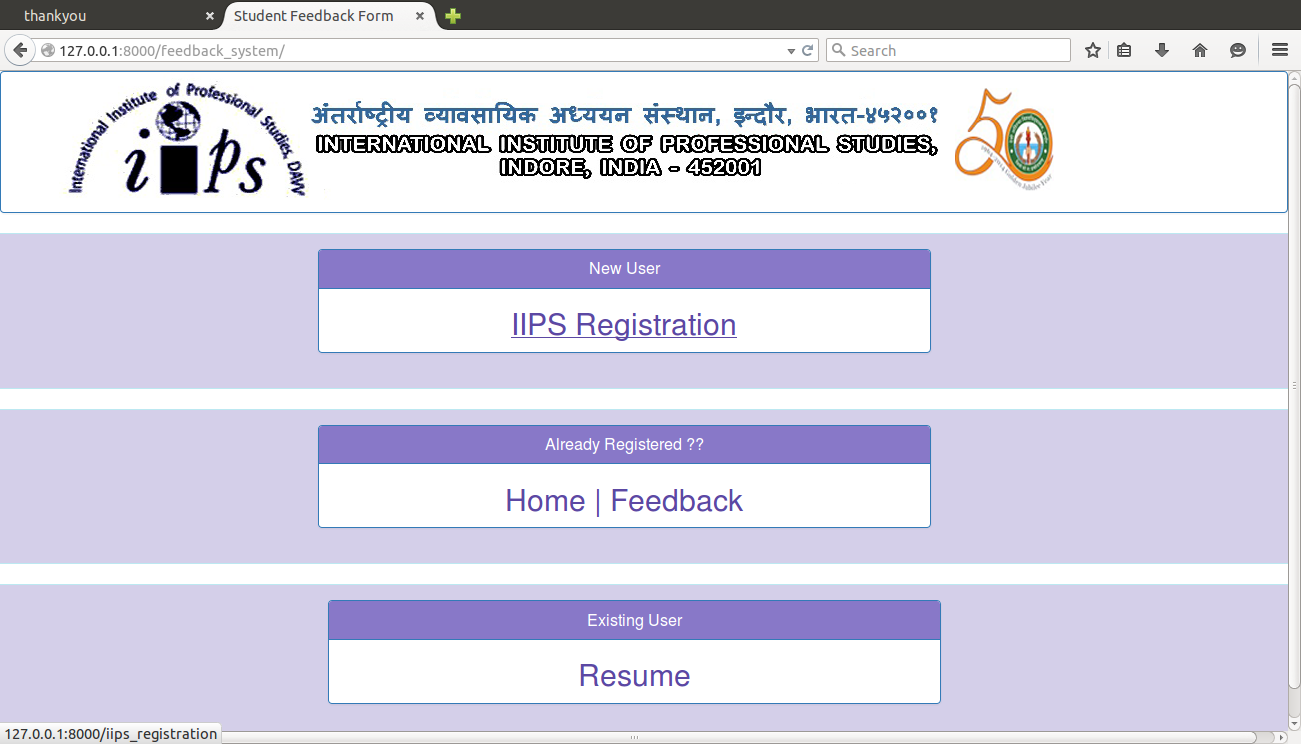
Django(python) is used as the user interface, covering the points:-

* Proper checks generated where field can not be left empty, like name, course, course year.
* Every ID is generated by system so eliminating confusion, like same name.
* Extra protection for field provided, like no name can have special characters, and no number field can have other than numbers.
* At the time of event date can not be earlier one than today.
* Dates are entered by default at many forms eliminating human errors.
* Fine calculation is late by date is calculated by system,so no errors.
* Form run in a flow of data so not difficult to understand.
* All events are click driven so not complex.

User interface forms are as follow:-

**Interfaces**

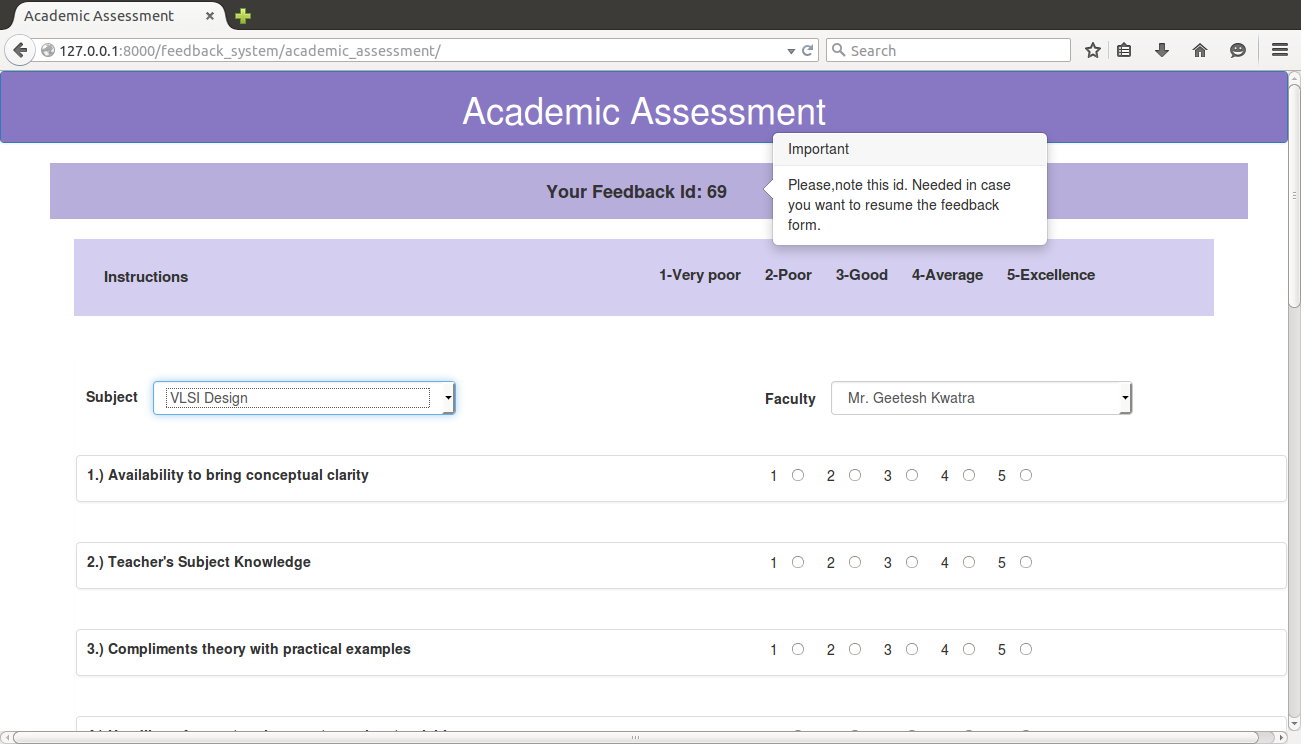
1. Home page



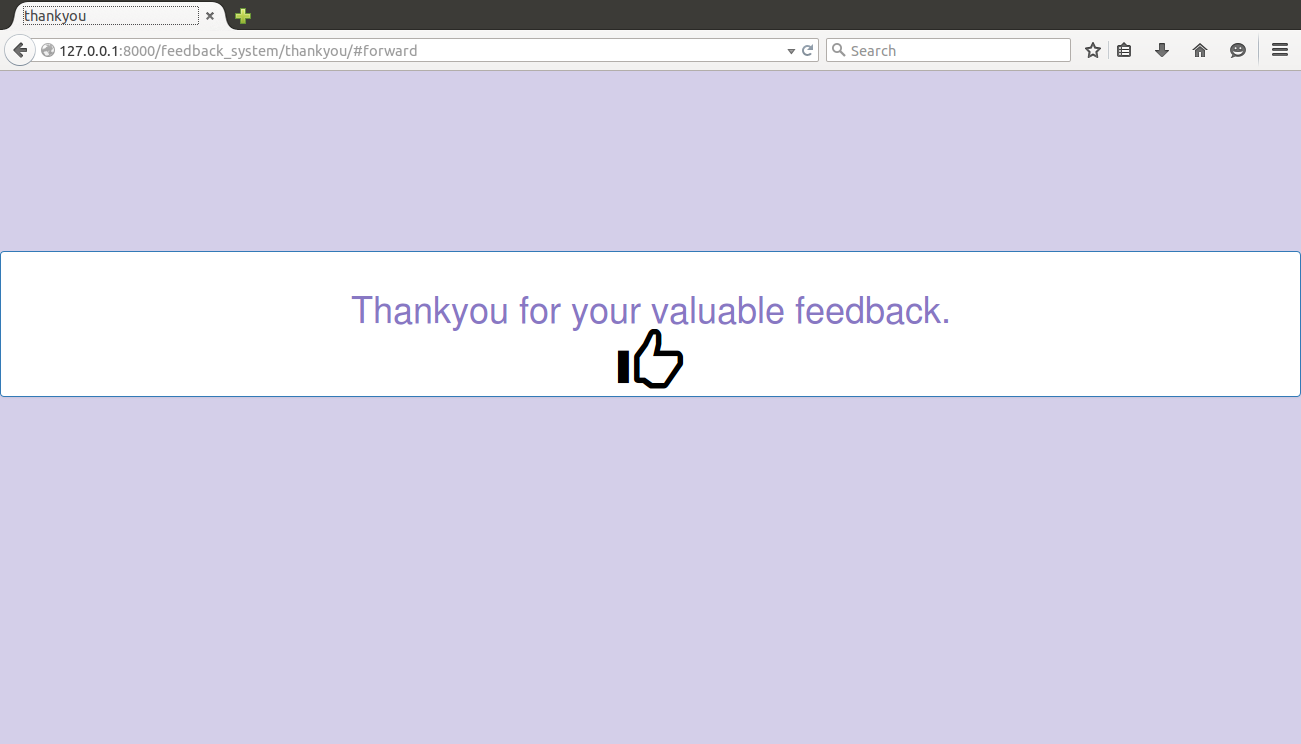
1. Infrastructure\_support page



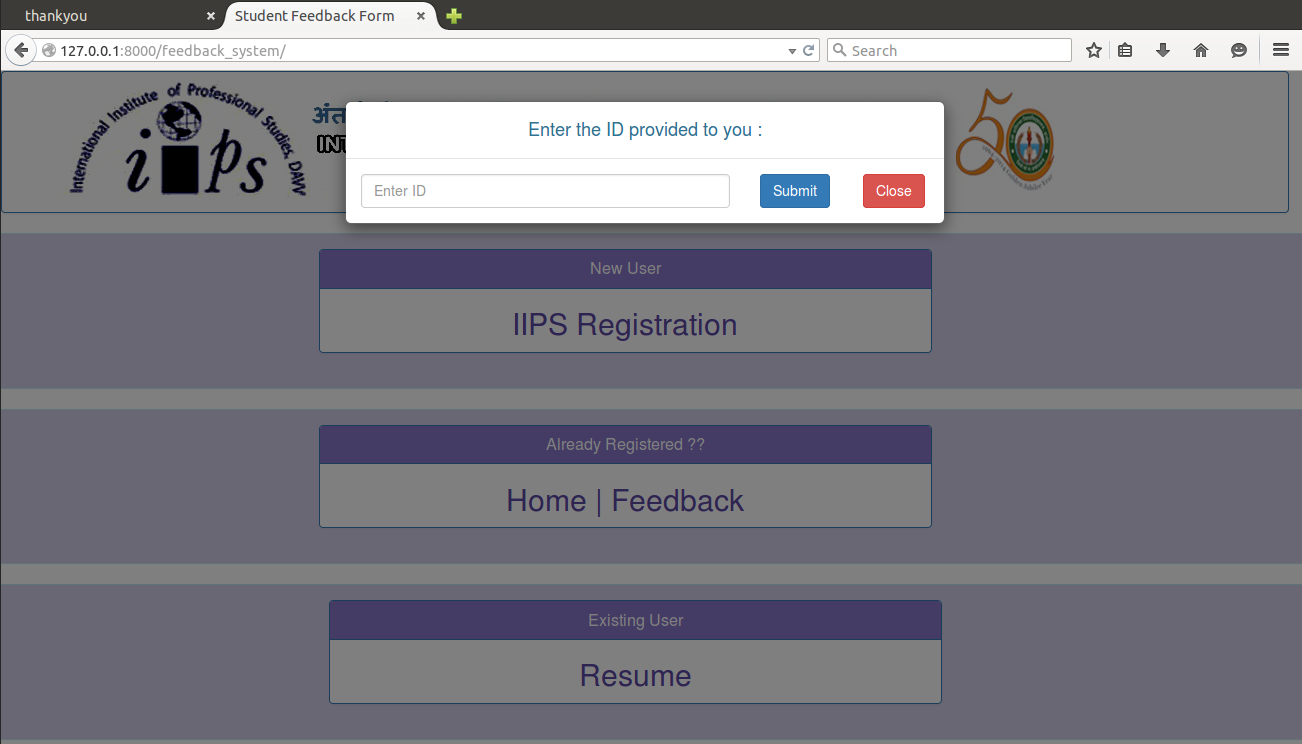
1. Academic assessment page



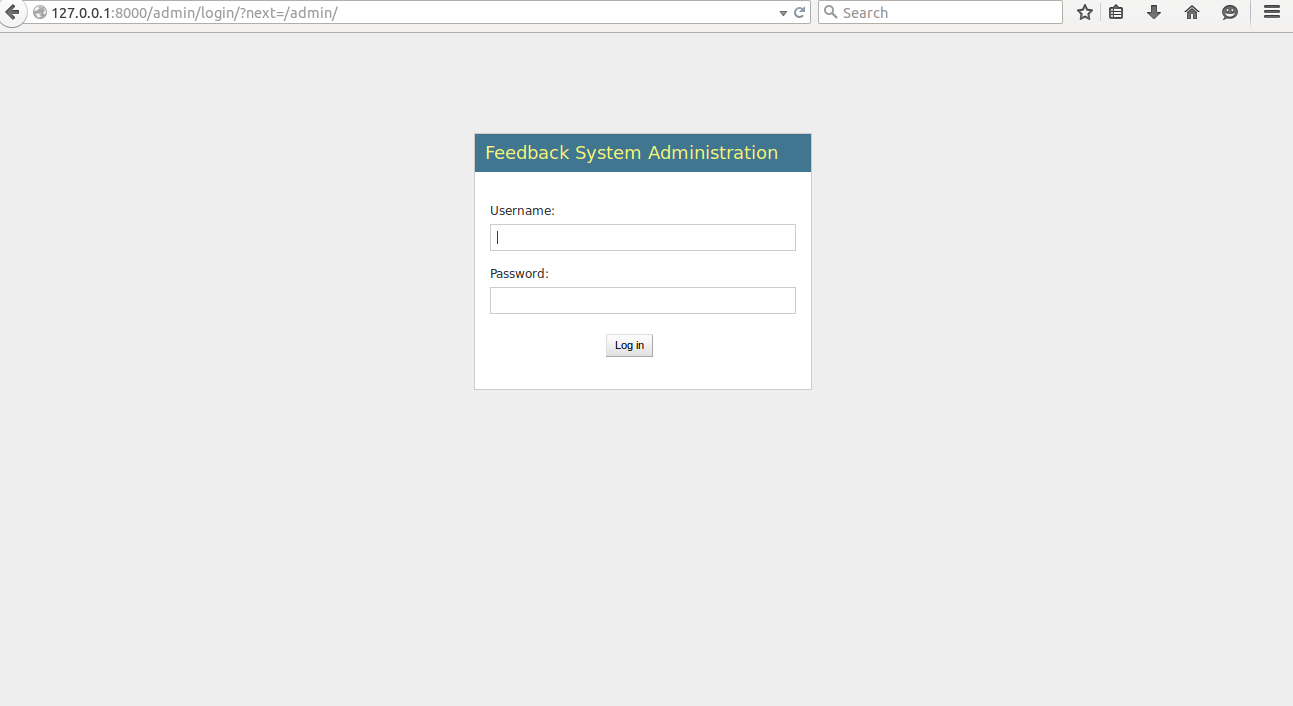
1. Thankyou page



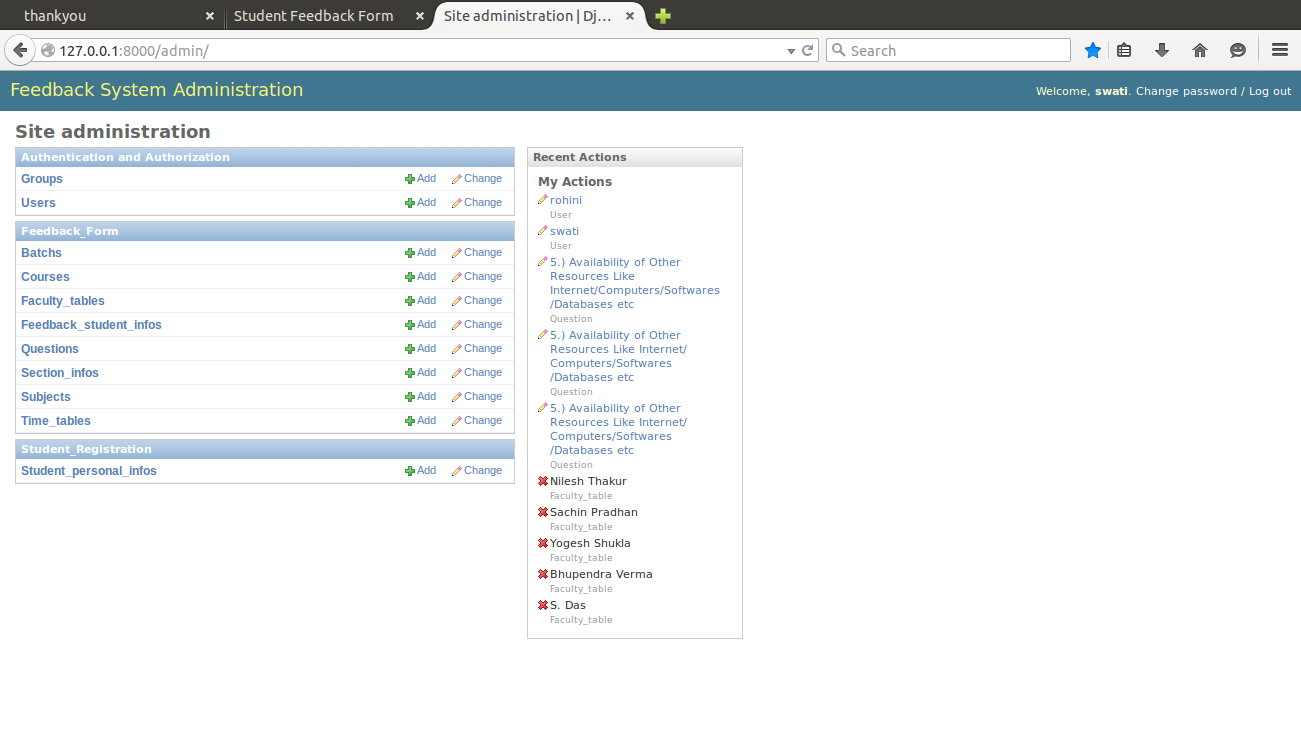
1. Resume page



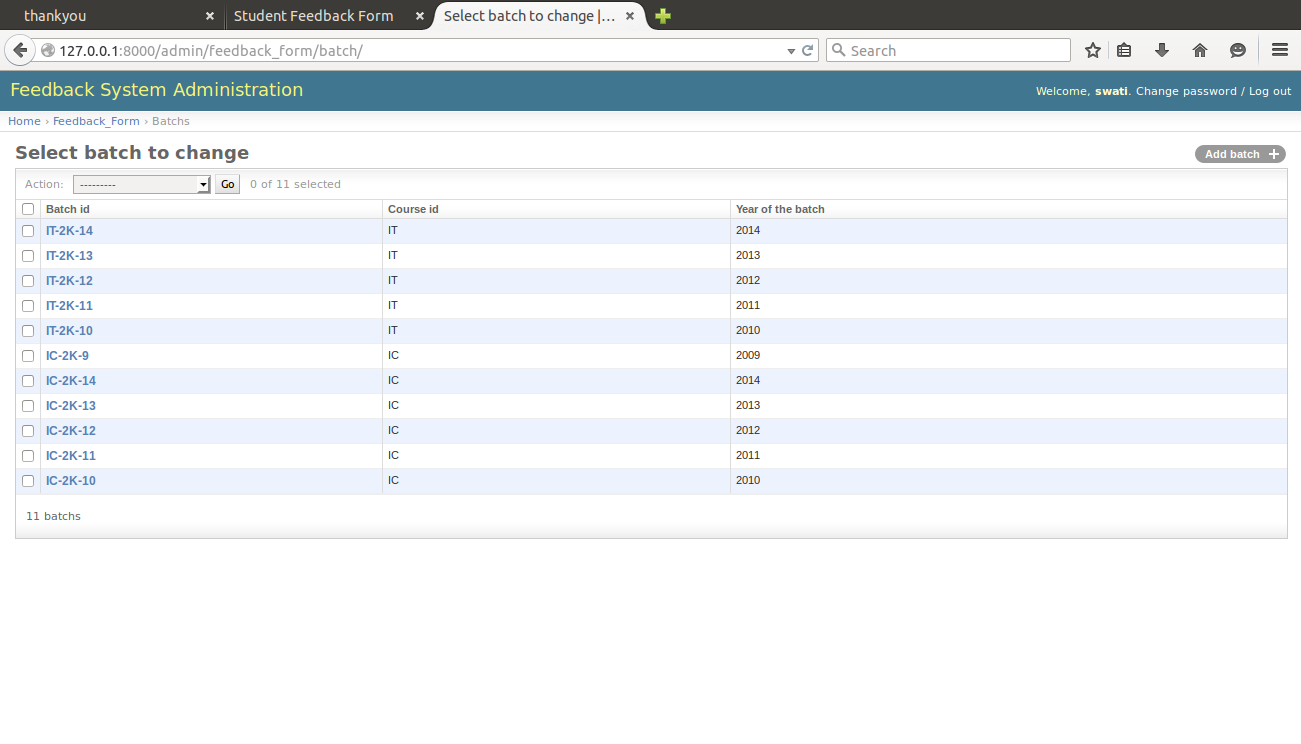
1. Admin page



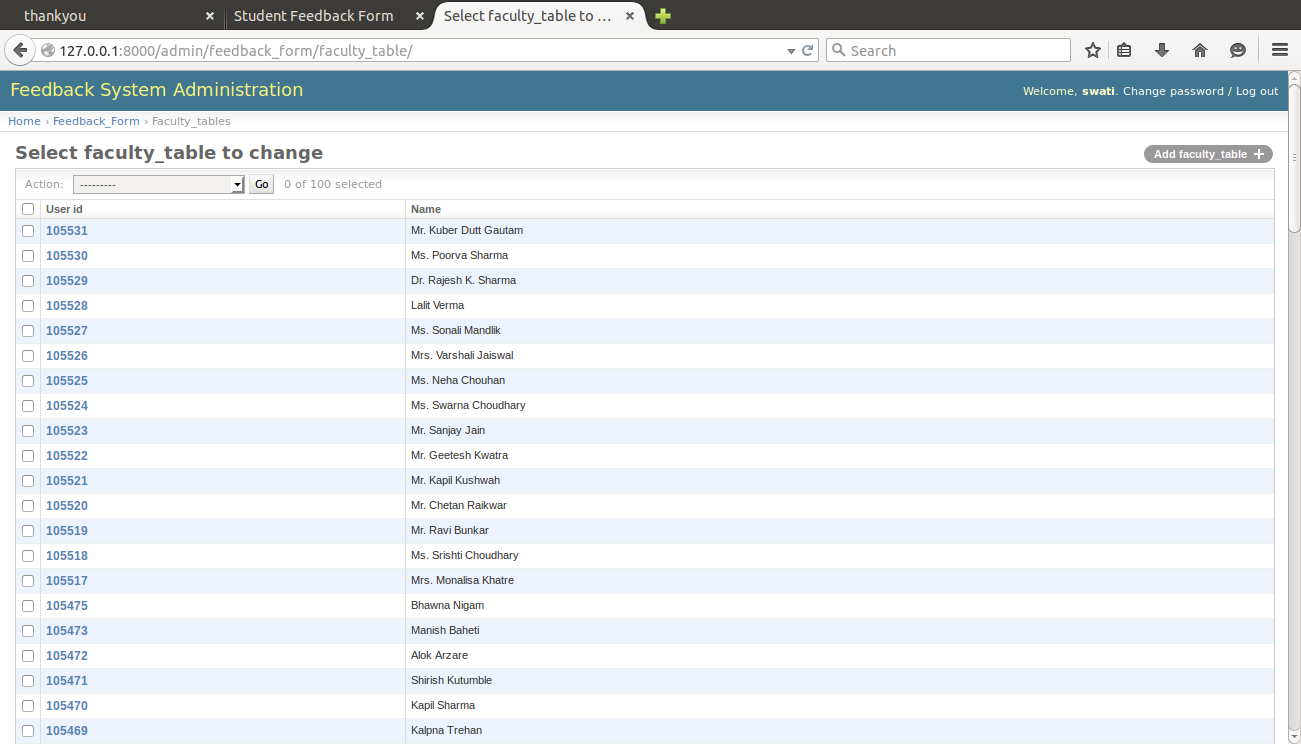
1. Admin home page



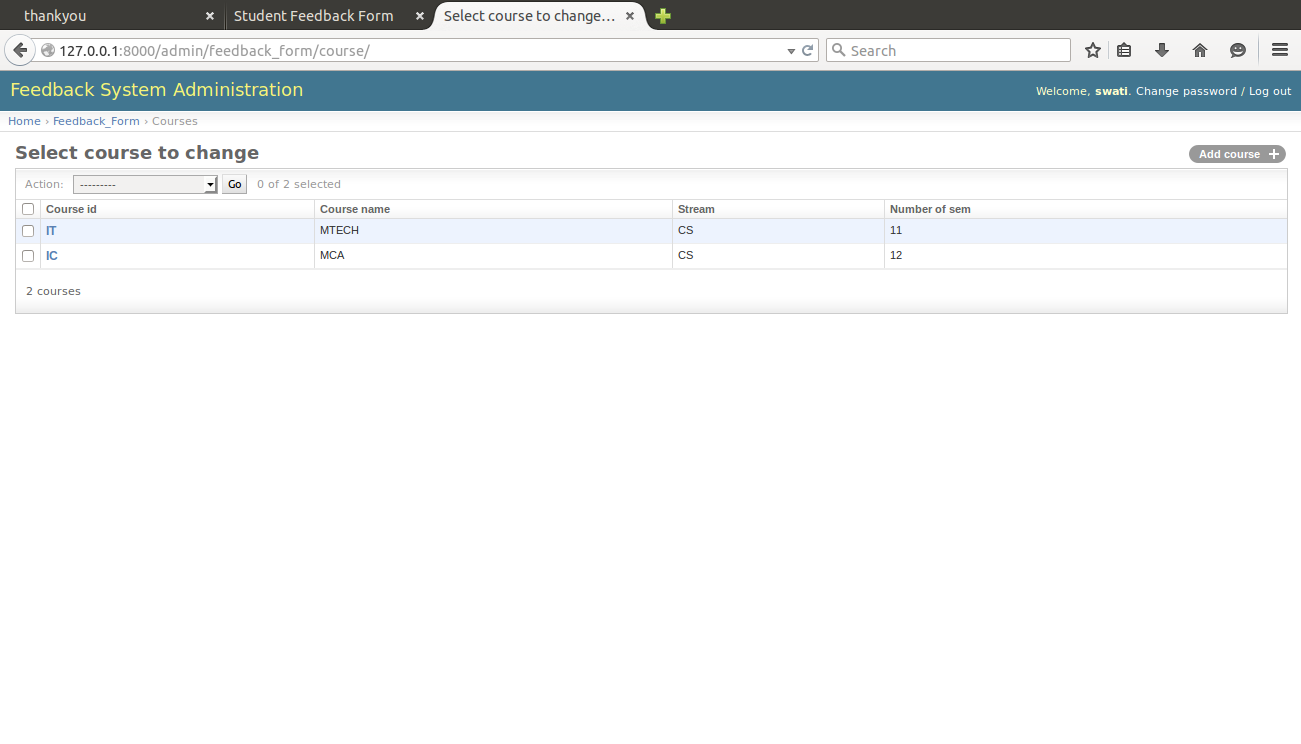
1. CRUD in Batch



1. CRUD in Faculty page



1. CRUD in Course page



**7. IMPLEMENTATION**

**IMPLEMENTATION**



**IMPLEMENTATION**

Implementation is the process of converting the new designed system into the operational one. There are various types of implementation. Some of these are as follows:

* Implementation of a computer system to replace a manual system.
* Implementation of a new computer system to replace an existing system.
* Implementation of a modified application to replace an existing one.

We are following the first implementation method to implement our system. In implementing our system, we do not need to give very hard user training.

The new system will be implemented fully after sometime on the basis of feedback given by the user.

**8. CODE SNIPPET**

**Views.py**

from django.shortcuts import render, get\_object\_or\_404, render\_to\_response

from feedback\_form.models import feedback\_student\_info

from django.http import HttpResponse, HttpResponseRedirect

from django.core.urlresolvers import reverse

from django.utils import timezone

from django import forms

from feedback\_form.models import \*

import json

import feedback\_form.models

from django.core import serializers

from django.template import RequestContext, loader, context

from feedback\_form.forms import loginForm

from feedback\_form.models import feedback\_student\_info

from django.template.response import TemplateResponse

def index(request):

return render(request, 'feedback\_form/index.html')

def action(request):

if ('id' and 'course' and 'semester' and 'course\_id' and 'batch\_id' and 'section') in request.POST:

q = feedback\_student\_info(batch\_id = request.POST['batch\_id'],course = request.POST['course'], semester = request.POST['semester'][:2], section = request.POST['section'], feedback\_session = timezone.now().year)

q.save()

else:

message = 'You submitted an empty form.'

return HttpResponse(message)

def login(request):

if request.method == 'POST' and request.is\_ajax:

form = loginForm(request.POST)

sem = request.POST.get('semester')

form.fields['semester'].choices = [(sem, sem)]

b = request.POST.get('batch')

form.fields['batch'].choices = [(b, b)]

s = request.POST.get('section')

form.fields['section'].choices = [(s, s)]

if form.is\_valid():

current\_id = Student\_unique\_id.objects.values\_list('unique\_id', flat=True).filter(unique\_id = request.POST['cur\_id'])

if int(request.POST['cur\_id']) in current\_id:

Student\_unique\_id.objects.filter(unique\_id = request.POST['cur\_id']).update(current\_sem = request.POST['semester'], current\_year = timezone.now().year)

q = feedback\_student\_info(batch\_id = request.POST['batch'],course = request.POST['programme'], semester = request.POST['semester'], section = request.POST['section'], feedback\_session = timezone.now().year)

q.save()

''' --- sessions of the variables are maintained here --- '''

request.session['fs\_id'] = q.fs\_id

request.session['course\_name'] = q.course

request.session['semester'] = q.semester

if (request.POST['section'] == '0'):

request.session['section'] = ''

else:

request.session['section'] = q.section

request.session['course\_id'] = q.batch\_id[:2]

request.session['batch\_id'] = q.batch\_id

return HttpResponseRedirect('/infrastructure\_support/')

else:

form = loginForm()

return render(request, 'feedback\_form/loginForm.html', {'form': form})

def thanks(request):

del request.session['fs\_id']

del request.session['course\_name']

del request.session['semester']

del request.session['section']

del request.session['course\_id']

del request.session['batch\_id']

return render(request, 'feedback\_form/thankyou.html')

def get\_batch(request, c\_id):

current\_course = feedback\_form.models.course.objects.get(course\_name=c\_id)

batches = feedback\_form.models.batch.objects.all().filter(course\_id=current\_course)

no\_of\_sem = feedback\_form.models.course.objects.values\_list('number\_of\_sem', flat=True).filter(course\_id = current\_course)

no\_of\_sem = int(no\_of\_sem[0])

batch\_dict = {}

for batch in batches:

batch\_dict[batch.batch\_id] = batch.batch\_id

sem = {}

sem[no\_of\_sem] = no\_of\_sem

data = [batch\_dict, no\_of\_sem]

return HttpResponse(json.dumps(data))

def get\_section(request, current\_batch):

batches\_with\_section = section\_info.objects.values\_list('batch\_id', flat = True)

section\_list = []

if current\_batch in batches\_with\_section:

available\_section = section\_info.objects.filter(batch\_id = current\_batch)

for i in available\_section:

section\_list.append(i.section)

return HttpResponse(json.dumps(section\_list))

def infrastructure\_support(request):

infrastructure\_qlist = Question.objects.filter(type = 'infrastructure support')

std\_id = request.session['fs\_id']

context = {'infrastructure\_qlist':infrastructure\_qlist, 'std\_id':std\_id}

return render(request, 'feedback\_form/infrastructure\_support\_info.html', context)

def infrastructure\_action(request):

if ('1' and '2' and '3' and '4' and '5' and '6') in request.POST:

''' --- fetching the session variables --- '''

if 'fs\_id' in request.session:

std\_id = request.session['fs\_id']

q=feedback\_student\_info.objects.get(pk=std\_id)

q.infrastructure\_support\_info\_set.create(books\_availability=request.POST['1'], basic\_requirements=request.POST['2'], technological\_support=request.POST['3'], study\_material=request.POST['4'], resourse\_availability=request.POST['5'], cleaniliness\_of\_class=request.POST['6'])

return HttpResponseRedirect('/academic\_assessment/')

else:

return HttpResponse('You submitted an empty form.')

def academic\_assessment(request):

''' fetching the data from the session variables '''

courseId = request.session['course\_id']

sem = request.session['semester']

batch = request.session['batch\_id']

section = request.session['section']

std\_id = request.session['fs\_id']

''' --- query to fetch data from the model ---'''

subject\_list = subject.objects.filter(course\_id = courseId, semester = sem, is\_viva\_or\_lab=0).exclude(pk\_\_in=academic\_assessment\_info.objects.values\_list('subject\_id', flat = True).filter(fs\_id = std\_id))

filled\_subject\_list = academic\_assessment\_info.objects.values\_list('subject\_id', flat = True).filter(fs\_id = std\_id)

faculty\_id\_list = time\_table.objects.values\_list('faculty\_id', flat = True).filter(batch\_id = batch, section = section)

faculty\_name\_list = []

for f in faculty\_id\_list:

faculty\_name\_list.append(faculty\_table.objects.filter(user\_id = f))

faculty\_qlist = Question.objects.filter(type = 'faculty assesment')

course\_qlist = Question.objects.filter(type = 'course assessment')

comment = Question.objects.filter(type = 'subject comment')

''' ----- maintaining session of subject\_list into dictionary ----- '''

subject\_list\_session = {}

context = {'subject\_list': subject\_list, 'faculty\_qlist': faculty\_qlist, 'faculty\_name\_list': faculty\_name\_list, 'course\_qlist': course\_qlist, 'comment': comment, 'std\_id':std\_id}

return render(request, 'feedback\_form/academic\_assessment\_info.html', context)

def academic\_action(request):

''' ~~~~~~~~~~ Accessing the session elements ~~~~~~~~~~~ '''

no\_of\_subject = request.session['total\_subject']

faculty\_id = faculty\_table.objects.values\_list('user\_id', flat = True).filter(name = request.POST['faculty'])

current\_faculty\_id = int(faculty\_id[0])

if ('subject' and '7' and '8' and '9' and '10' and '11' and '12' and '13' and '14' and '15' and '16' and 'comment1' and '18' and '19' and 'comment2') in request.POST:

if 'fs\_id' in request.session:

std\_id = request.session['fs\_id']

a = feedback\_student\_info.objects.get(pk=std\_id)

a.academic\_assessment\_info\_set.create(subject\_id = request.POST['subject'], faculty\_id = current\_faculty\_id, conceptual\_clarity = request.POST['7'], subject\_knowledge = request.POST['8'], practical\_example = request.POST['9'], handling\_capability = request.POST['10'], motivation = request.POST['11'], control\_ability = request.POST['12'], course\_completion = request.POST['13'], communication\_skill = request.POST['14'], regularity\_punctuality = request.POST['15'], outside\_guidance = request.POST['16'], syllabus\_industry\_relevance = request.POST['18'], sufficiency\_of\_course = request.POST['19'], suggestion\_for\_subject = request.POST['comment1'], suggestion\_for\_course = request.POST['comment2'])

if no\_of\_subject == 1:

return HttpResponseRedirect('/thankyou/')

else:

return HttpResponseRedirect('/academic\_assessment/')

else:

return HttpResponse('You submitted an empty form.')

def get\_faculty\_name(request, sub\_name):

urrent\_course\_id = request.session['course\_id']

current\_section = request.session['section']

if current\_section == '0':

current\_section = ''

current\_faculty\_id = time\_table.objects.values\_list('faculty\_id', flat = True).filter(subject\_id = sub\_name, section = current\_section)

faculty\_name = faculty\_table.objects.filter(user\_id = current\_faculty\_id)

data = []

for f in faculty\_name:

data = [f.name]

return HttpResponse(json.dumps(data))

def resume\_action(request):

if 'fs\_id' in request.POST:

is\_present = feedback\_student\_info.objects.filter(pk = request.POST['fs\_id'])

if (len(is\_present) > 0 ):

query\_value = feedback\_student\_info.objects.get(pk = request.POST['fs\_id'])

request.session['fs\_id'] = query\_value.fs\_id

request.session['course\_name'] = query\_value.course

request.session['semester'] = query\_value.semester

request.session['section'] = query\_value.section

request.session['course\_id'] = query\_value.batch\_id[:2]

request.session['batch\_id'] = query\_value.batch\_id

infrastructure\_presence = infrastructure\_support\_info.objects.filter(fs\_id = is\_present)

if (len(infrastructure\_presence) > 0 ):

''' fetching the data from the session variables '''

courseId = request.session['course\_id']

sem = request.session['semester']

batch = request.session['batch\_id']

std\_id = request.session['fs\_id']

subject\_list = subject.objects.filter(course\_id = courseId, semester = sem, is\_viva\_or\_lab=0).exclude(pk\_\_in=academic\_assessment\_info.objects.values\_list('subject\_id', flat = True).filter(fs\_id = std\_id))

if len(subject\_list) == 0:

return HttpResponse("Your feedback form has been submitted successfully.")

else:

return HttpResponseRedirect('/academic\_assessment/')

else:

return HttpResponseRedirect('/infrastructure\_support/')

else:

return HttpResponse("Oops! You are not a Existing User. Please <a href='../login' > <b>CLICK</b> </a> here to proivide your valuable feedback.")

**9. TESTING**

**TESTING**



Testing should be done throughout the implementation process. Even before an application is installed, it makes sense to verify that the basic platform is capable of achieving its design capabilities. System testing is a critical process. Testing is a process of executing a program with the explicit intention of finding errors that is, making the program fail. This help in finding the bottlenecks in the system. Executing a program in a simulated environment performs testing. The feedback from testing phase generally produces changes in the software to deal with errors and failures that are uncovered.

**9. 1 Black box testing:**

In Black Box testing or functional testing test cases are decided. Test cases are decided on the basis of the requirements or the specifications of the program or module.

Black Box testing is done in the project to remove the errors:

* Incorrect or missing function.
* Interface errors.
* Errors in data structures or external database access.
* Behavioural or performance error.
* Errors in initiation & termination.

**9.2 White Box testing:**

The White Box testing or Structural testing performs close operation of procedural details. They test the software logical path by having test cases exercising specific sets of conditions and loops.

White Box testing is done in the project to remove the errors.

* All modules path have been exercised at least once.
* Exercised on logical decisions.
* Executed all loops at their boundaries and within their operational bounds.
* Exercised internal data structure to ensure their validity.

**9.3 Unit Testing:**

Unit testing focuses on verification efforts of the smallest grid of software designing i.e. a software component or module is tested. This testing is done at the coding phase. This testing uses procedural design as guide to test major controls path and uncovers errors within the module boundary.

Following tests were performed during unit test:

1. **Module Interface Testing:**

Module interface was tested to ensure information flow in and out of the program unit.

2. **Local Data Structure Testing:**

Local Data Structure were tested to ensure that data store temporally maintain their integrity during all steps in algorithm execution.

3. **Boundary Condition Test:**

Boundary Conditions were tested to make sure that the modules operate properly at boundaries.

4. **Independent Path Test:**

All independent pathsthrough control structure were checked to make sure that all statements in a module have been executed.

5. **Error Handling Path Test:**

Error handling path test was performed to handle exceptions.

**9.4 Integration Testing:**

Integration testing is done to tackle the problem of interface i.e. putting all the interfaces together. When the separate modules are put together and work in an integrated manner, this testing is performed. This testing is a Systematic technique. This testing is performed to check that data should not be lost across an interface. The objective is to take unit tested modules and build a program structure that has been dictated by design.

Regression**:** Regression was done to ensure proper working of each module with the whole system. Each module was embedded in the system and the whole tested for integrity.

**9.5 System Testing:**

System testing is done when the entire system has been fully integrated. The purpose of the system testing is to test how the different modules interact with each other and whether the system provides the functionality that was expected.

Security**:** Security was added to the system by making it password protected.

**10. FUTURE PROSPECTIVE**

**FUTURE PROSPECTIVE**



The project can be further expanding to many level as:-

* Analysis module can be integrated with the project itself.
* Each Student can get/informed about unique id via mail.
* Project link will be activated for particular days.
* Student will get reminder about feedback submission via mail.
* Confirmation of mail-id functionality can be added.
* Confirmation of successful submission of feedback can be provided via mail.

**BIBLIOGRAPHY**



[01] Elias M.Awad, System Analysis and Design, IInd edition 2009, Galgotia Publication Pvt. Ltd, New Delhi.

[02] V.Rajaraman, Analysis and Design of Information System, IInd Edition 2002, Prentice Hall of India, New Delhi.

[03] Roger S Pressman, Software Engineering: A Practitioner's Approach, VIth edition 2005, Tata McGraw- Hill Publishing Company Limited, New Delhi.

[04] [Pankaj](http://www.cse.iitk.ac.in/users/jalote) [Jalote HYPERLINK "http://www.cse.iitk.ac.in/users/jalote",](http://www.cse.iitk.ac.in/users/jalote) An Integrated Approach to Software Engineering**,** IIIrd Edition, [Narosa](http://www.narosa.com/) [Publishing](http://www.narosa.com/) [House](http://www.narosa.com/) **(**for India).

**REFERENCES**



1. <http://djangoproject.com>
2. <http://mysql.com>
3. <http://python.org>
4. <http://crispyform.com>
5. <http://apache.com>