Project Report

Enigma
Tanuj Kaza
Harsh Bansal
Bhavya Bahl

05/11/16

Abstract

Feeder was started by a group of IIT Bombay sophomores with a view to changing how courses function in their institute(and elsewhere). It is meant to bridge the communication gap between the students and the professors which in turn will improve the quality of the course being dished out.

The instructors can receive constructive feedback on various aspects of the course and can thus dynamically change the ocurse structure for the better. The students too are sent reminders for the assignments and deadlines so that they do not miss out on important submissions and can plan their schedule more effectively.

I. Design

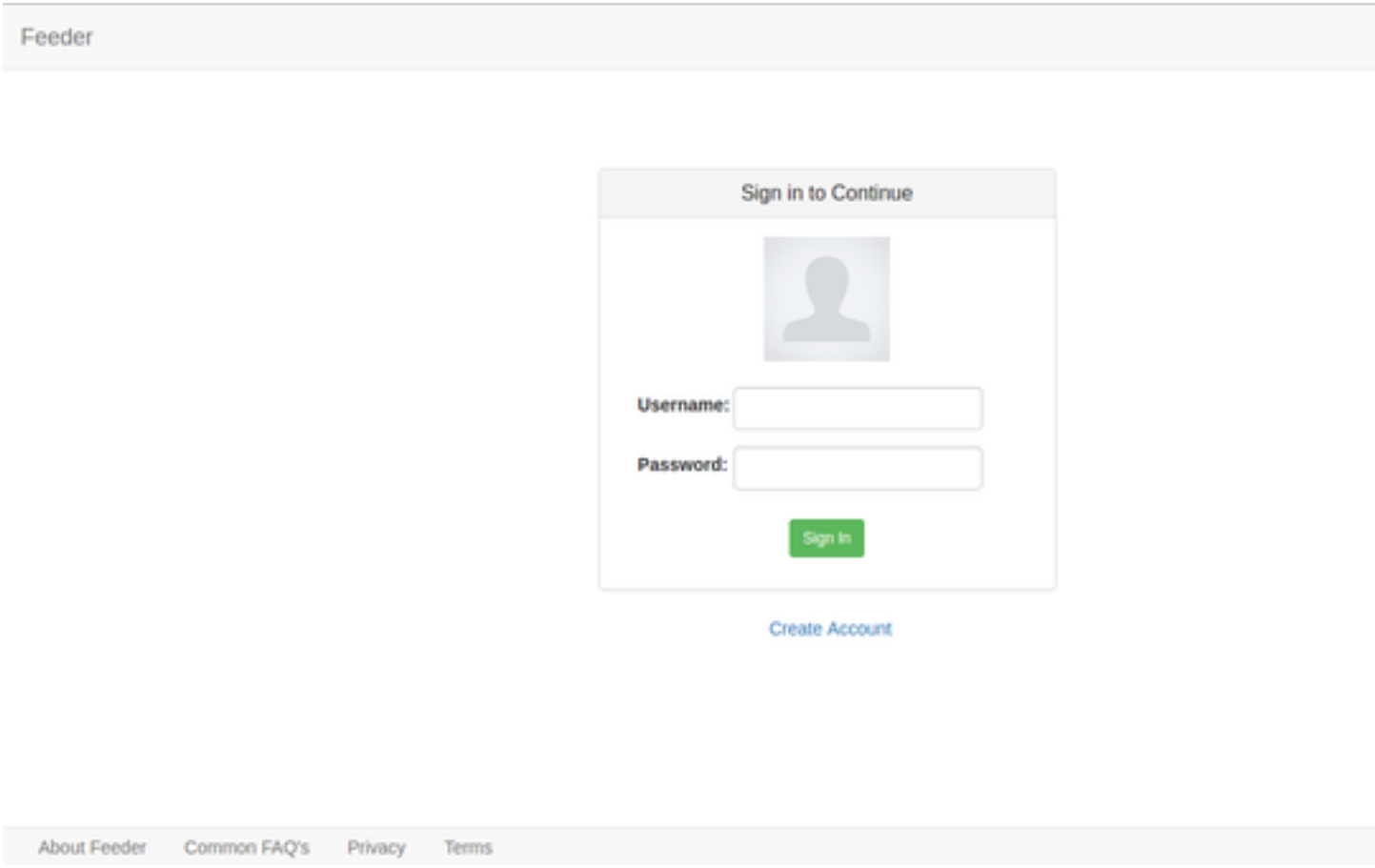
A. The project is built on a Django Framework

B. The student interface is an Android Application

C. The android application uses the django server as the backend server

D. The App uses Bootstrap for styling and for embellishing the user interface of the application

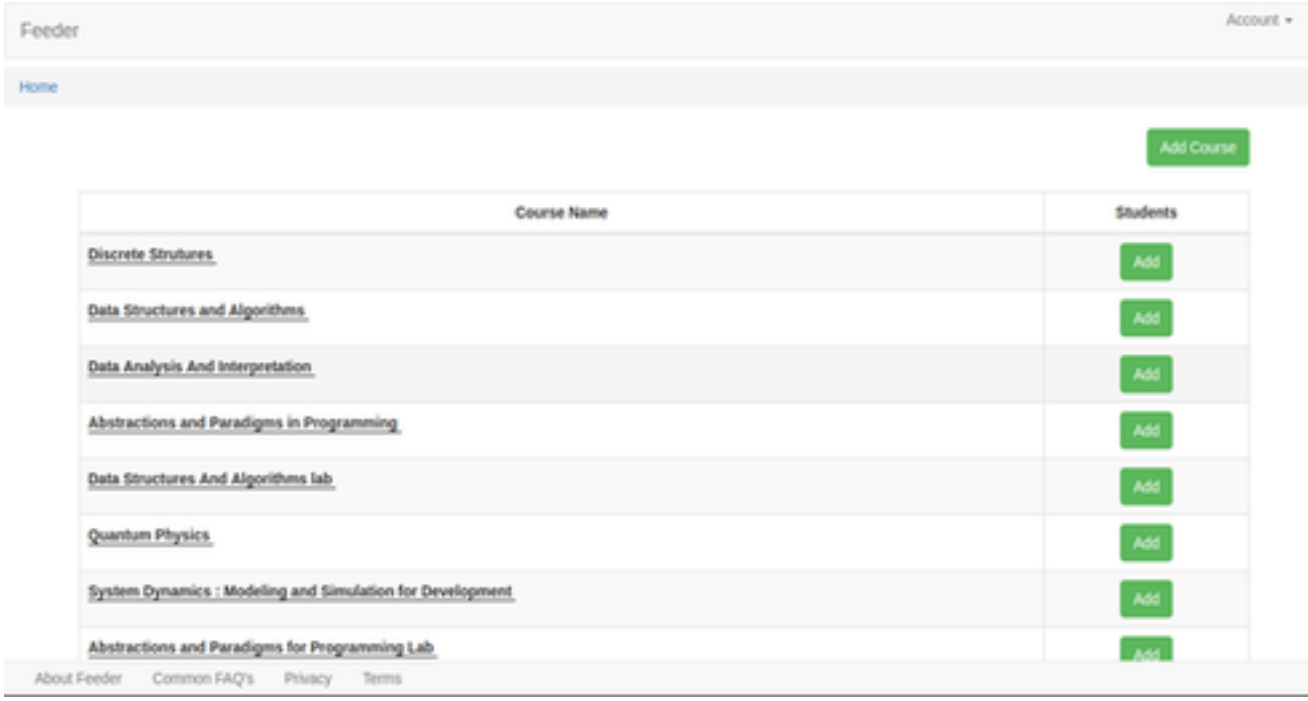
E. The ReportLab and Imaging Libraries of python are used for generating graphs

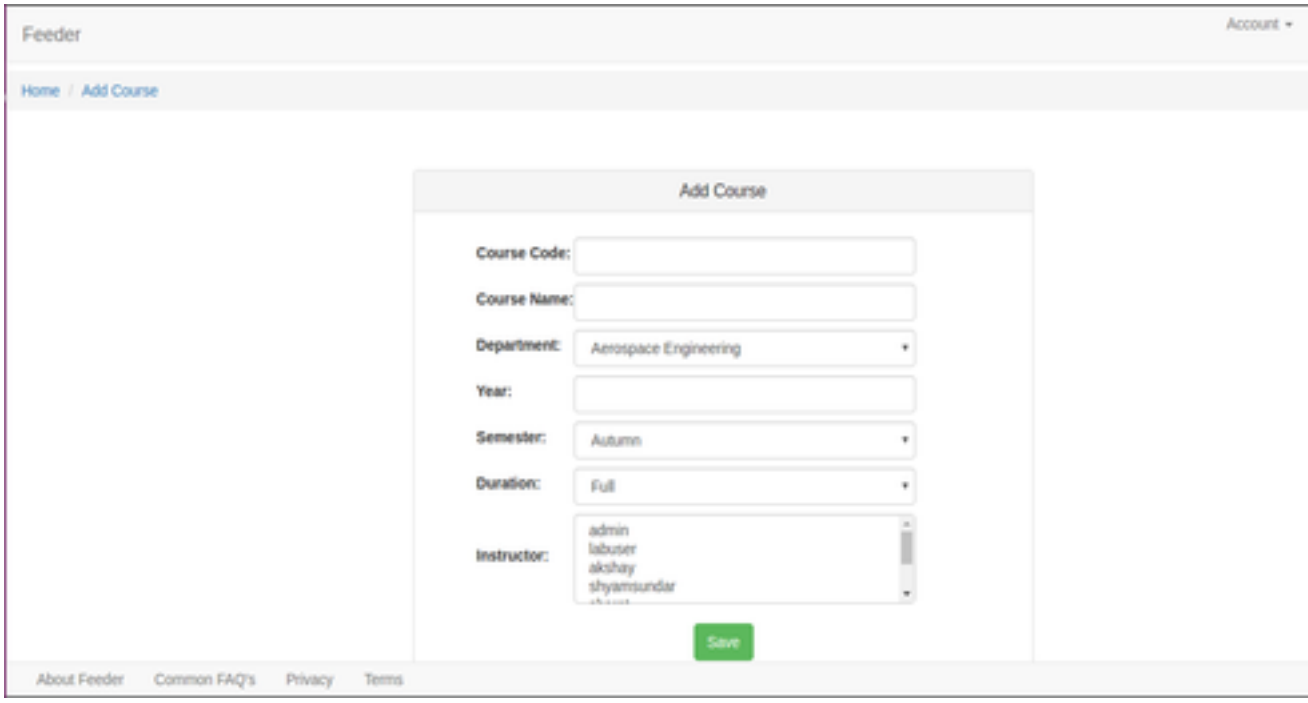


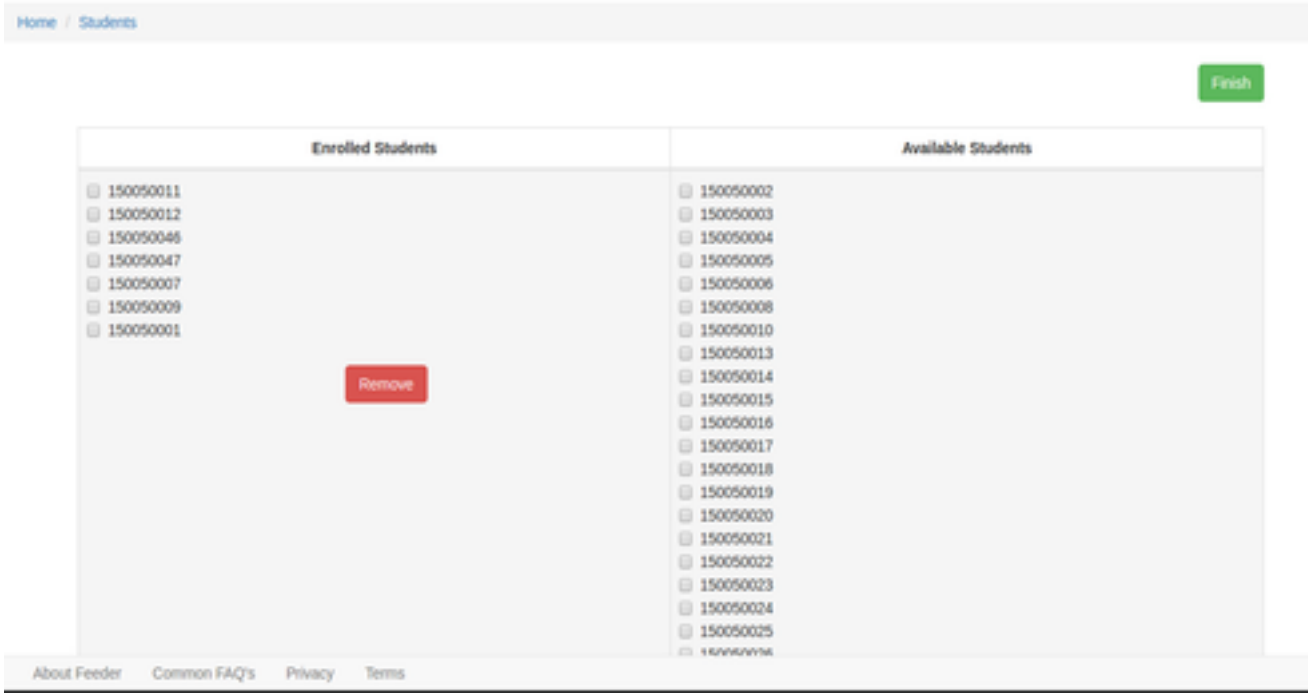
I. Features and Technical Details

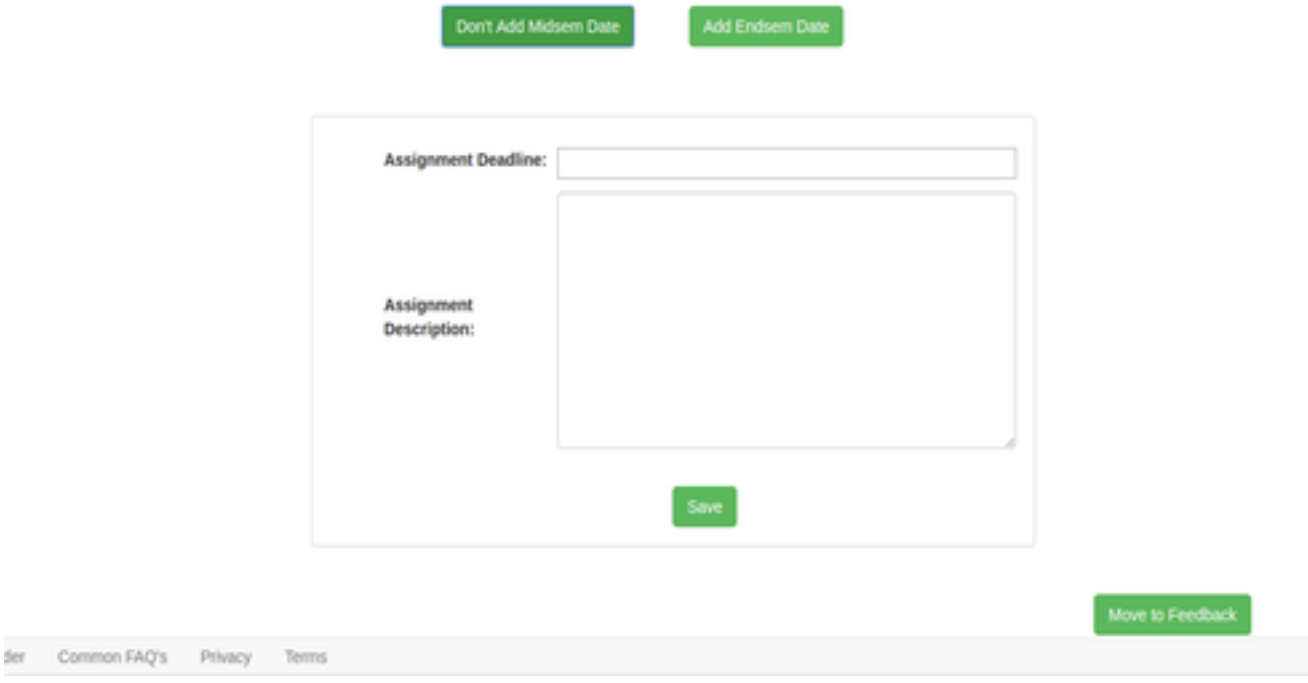
A. Admin Features

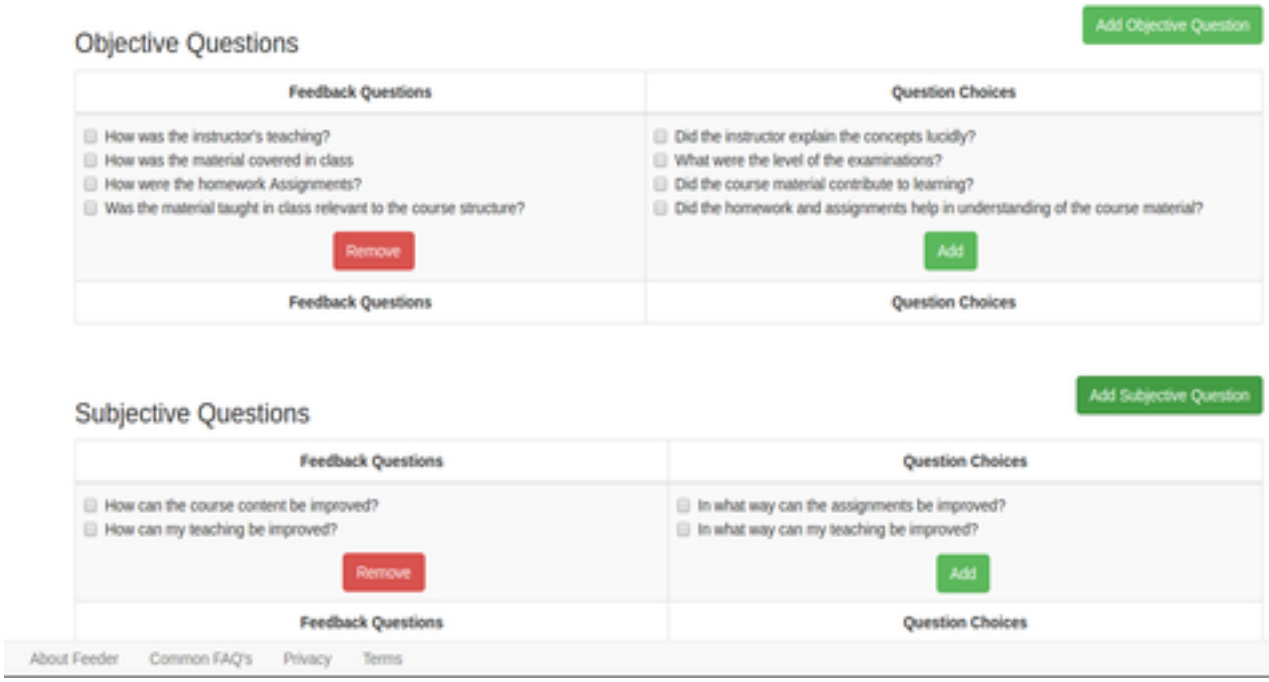
1. Admin can add students into courses from a list of enrolled students
2. The list of students is imported from a csv file named 'student.csv' when the admin logs in. If there are any new students in the csv file, then even they will be added to the database
3. The other functionality for the admin is adding courses. The fields in the add course form are mandatory and must be filled. The year field must be an integer(example : 2016). Apart from that Department is selected from the list of departments in IIT Bombay.
4. Now the instructors must be selected for this course. There is a many to many mapping from instructors to course. One course can have multiple instructors(eg.MA105 and other freshmen courses, or consider an instructor and the Teaching Assistants(TA's) for the course. The TA's are treated as instructors in this app.). So all the instructors taking this course must be selected form the multiple select list.
5. Following this, the admin is prompted for midsem and endsem deadlines(It is not compulsory that the admin adds a midsem or endsem deadline and hence functionality has been provided where he can move directly to the feedback forms)
6. Some courses like CS101 in first year have multiple endsem and midsem examinations(Lab and theory courses). Hence functionality has been provided for the admin to add multiple midsem and endsem deadlines.
7. The midsem and endsem deadline buttons are toggle buttons and can be clicked if you realise you do not want to add a deadline
8. In the feedback form, there are some default objective questions(Questions which will have answers in the form of ratings from 1:5) and some subjective questions(Subjective Text Questions).
9. In addition to that, the admin can remove some of these questions, and add some more default questions which have been provided. In addition to this, the admin can also add some of his own objective and subjective questions(Not required, but he has been provided the functionality!). After all questions have been added, he has to add a feedback deadline and then he progresses to the endsemester form.(Where similar functionality is provided)Thus we have customized Feedback Forms
10. Apart from this we have a privacy policy and a terms of service too[Links given in the footer](Source has been cited) to which we strictly adhere to :P

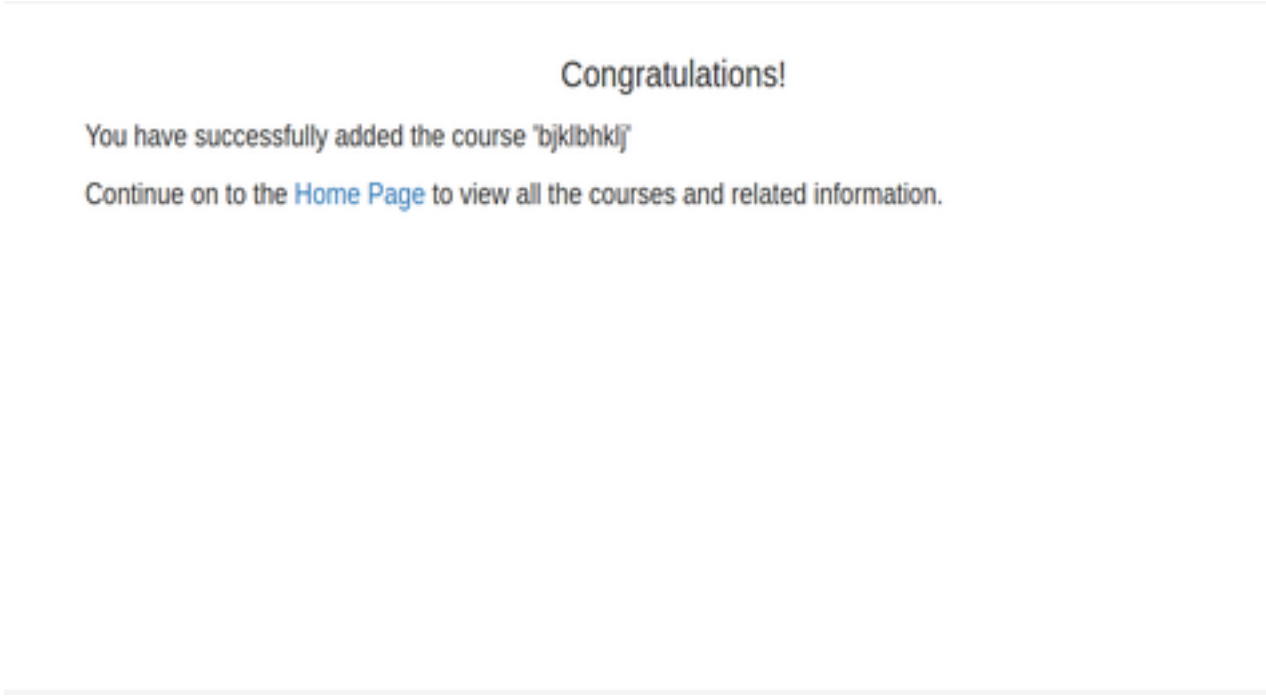

Feeder
Account
Home
Add Course
Course Name
Students
Discrete Strutures
Data Structures and Algorithms
Data Analysis And Interpretation
Abstractions and Paradigms in Programming
Data Structures And Algorithms lab
Quantum Physics
System Dynamics : Modeling and Simulation for Development
Abstractions and Paradigms for Programming Lab
Add
About Feeder
Common FAQ's
Privacy
Terms


Feeder
Account
Home / Add Course
Add Course
Course Code:
Course Name:
Department:
Aerospace Engineering
Year:
Semester:
Autumn
Duration:
Full
Instructor:
admin
labuser
akshay
shyamsundar
Save
About Feeder
Common FAQ's
Privacy
Terms


Home / Students
Finish
Enrolled Students
Available Students
150050011
150050012
150050046
150050047
150050007
150050009
150050001
Remove
150050002
150050003
150050004
150050005
150050006
150050008
150050010
150050013
150050014
150050015
150050016
150050017
150050018
150050019
150050020
150050021
150050022
150050023
150050024
150050025
About Feeder
Common FAQ's
Privacy
Terms


Don't Add Midsem Date
Add Endsem Date
Assignment Deadline:
Assignment Description:
Save
Move to Feedback
Common FAQ's
Privacy
Terms


Add Objective Question
Objective Questions
Feedback Questions
Question Choices
How was the instructor's teaching?
How was the material covered in class
How were the homework Assignments?
Was the material taught in class relevant to the course structure?
Remove
Did the instructor explain the concepts lucidly?
What were the level of the examinations?
Did the course material contribute to learning?
Did the homework and assignments help in understanding of the course material?
Add
Feedback Questions
Question Choices
Add Subjective Question
Subjective Questions
Feedback Questions
Question Choices
How can the course content be improved?
How can my teaching be improved?
Remove
In what way can the assignments be improved?
In what way can my teaching be improved?
Add
Feedback Questions
Question Choices
About Feeder
Common FAQ's
Privacy
Terms


Congratulations!
You have successfully added the course "bjklbhklj"
Continue on to the Home Page to view all the courses and related information.

B. Instructor Features

1. Once you login with any of the above mentioned credentials, there are a list of courses being taken by the instructor and the ability to add assignments and feedback forms in each of the courses. The courses can be viewed by only the instructor taking the course and the Teaching Assistants for that course
2. The assignments can be seen in the form of deadlines pending and those that have been completed, with the capability of editing the assignment fields(Deadlines,specifications.etc). The Feedback can also be seen on the same page. This makes for a compact application.
3. Assignments can be added into each course[Foreign Key Relationship] in real time with the provision for adding deadlines and specifications
4. The instructor can also view the feedback that has been submitted by the students, in interactive formats like graphs for objective questions(As well as in aggregate format) and the subjective results are also in aggregate format.We have used The ReportLab and the Imaging Libraries for showing the graphs
5. In the feedback form, there are some default objective questions(Questions which will have answers in the form of ratings from 1:5) and some subjective questions(Subjective Text Questions). This accounts for all types of questions
6. There is a many to many field relationship between feedback questions and feedback forms. There is a foreign key from feedback answers to feedback questions and feedback forms. Hence each answer is uniquely identified
7. In addition to that, the instructor can remove some of these questions, and add some more default questions which have been provided. In addition to this, the instructor can also add some of his own objective and subjective questions(Not required, but he has been provided the functionality!). After all questions have been added, he will be redirected to the course page.
8. The instructor can add as many feedback forms as desired.
9. We have provided facebook login for the instructors
10. You can click on the feeder brand name in the top left corner to be directed to the home page.

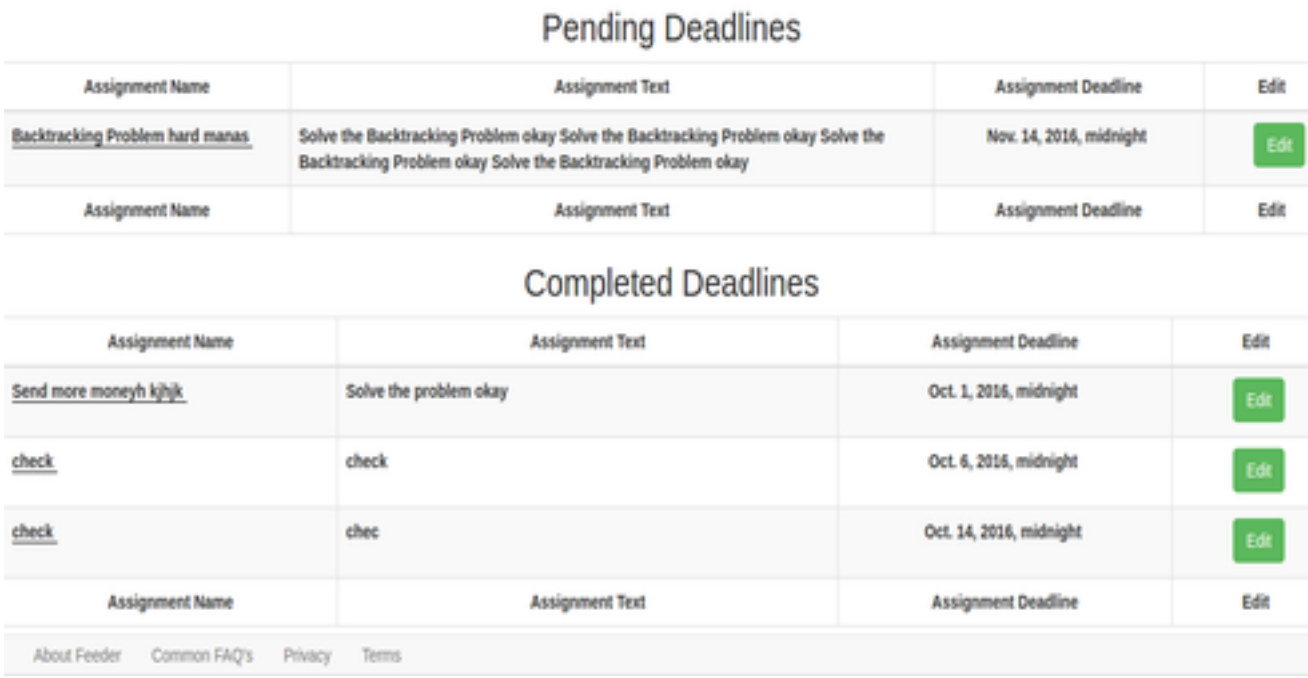
Course Name	Assignments		Feedback	
Discrete Strutures	View	Add	View	Add
Data Structures and Algorithms	View	Add	View	Add
Data Analysis And Interpretation	View	Add	View	Add
Abstractions and Paradigms in Programming	View	Add	View	Add
Data Structures And Algorithms lab	View	Add	View	Add
Abstractions and Paradigms for Programming Lab	View	Add	View	Add
Formal Verification	View	Add	View	Add
Check for course	View	Add	View	Add

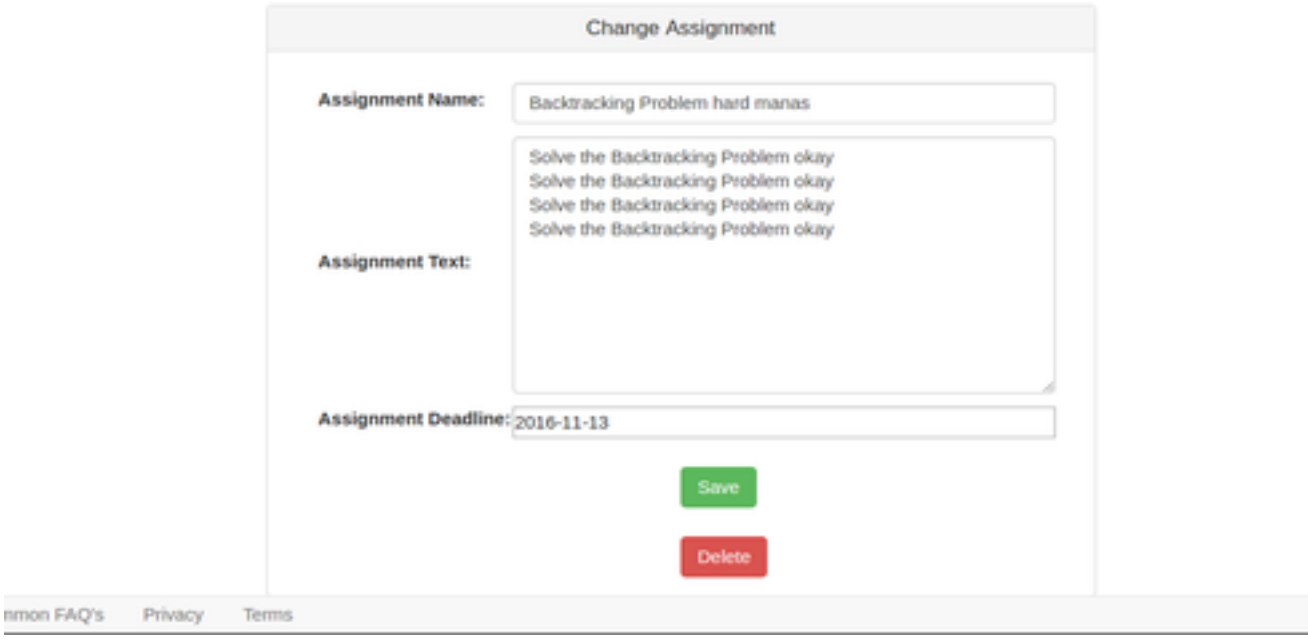
Objective Questions

Question	Poor	Fair	Good	Very Good	Excellent	Total Responses	Graphical Results
How was the instructor's teaching?	12	23	45	34	56	170	View
How was the material covered in class	14	34	23	56	45	172	View
How were the homework Assignments?	23	67	34	67	45	236	View
Was the material taught in class relevant to the course structure?	23	67	45	23	89	247	View
Did the instructor explain the concepts lucidly?						None	View
Question	Poor	Fair	Good	Very Good	Excellent	Total Responses	Graphical Results

Subjective Questions

Question	Results


Pending Deadlines
Assignment Name
Assignment Text
Assignment Deadline
Edit
Backtracking Problem hard manas
Solve the Backtracking Problem okay Solve the Backtracking Problem okay Solve the Backtracking Problem okay Solve the Backtracking Problem okay
Nov. 14, 2016, midnight
Edit
Assignment Name
Assignment Text
Assignment Deadline
Edit
Completed Deadlines
Assignment Name
Assignment Text
Assignment Deadline
Edit
Send more moneyh kjhjk
Solve the problem okay
Oct. 1, 2016, midnight
Edit
check
check
Oct. 6, 2016, midnight
Edit
check
chec
Oct. 14, 2016, midnight
Edit
Assignment Name
Assignment Text
Assignment Deadline
Edit
About Feeder
Common FAQ's
Privacy
Terms

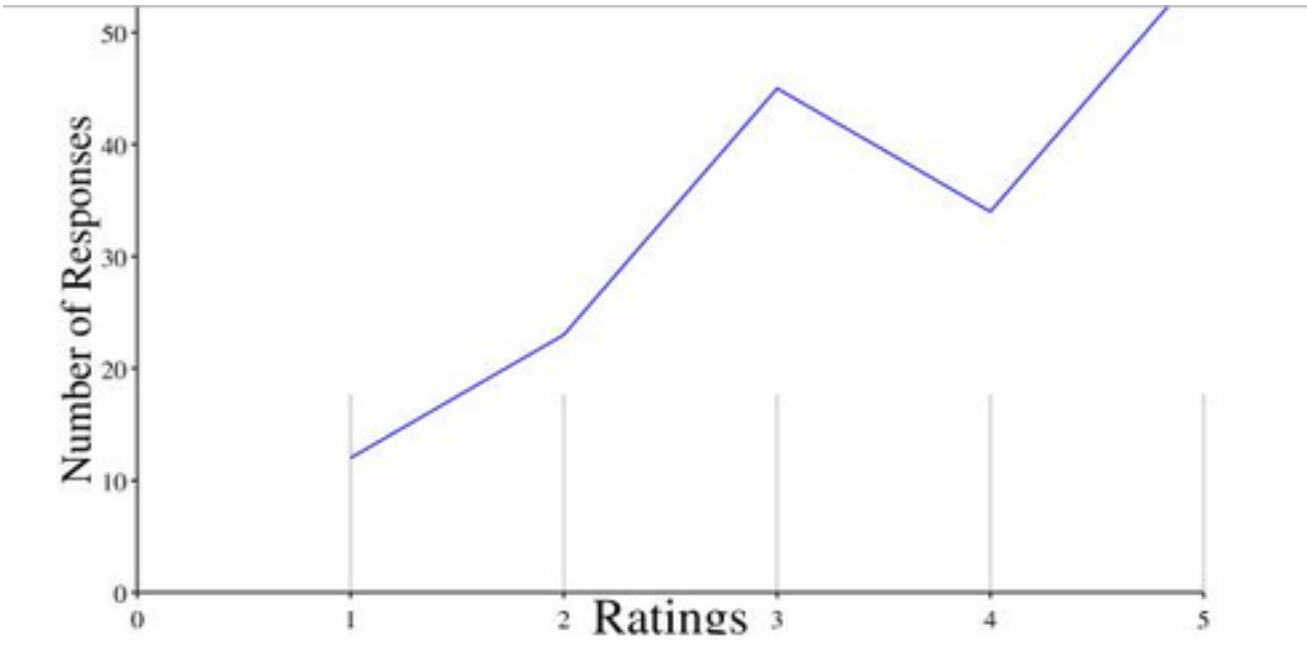

Change Assignment
Assignment Name:
Backtracking Problem hard manas
Assignment Text:
Solve the Backtracking Problem okay
Solve the Backtracking Problem okay
Solve the Backtracking Problem okay
Solve the Backtracking Problem okay
Assignment Deadline:
2016-11-13
Save
Delete
nmon FAQ's
Privacy
Terms

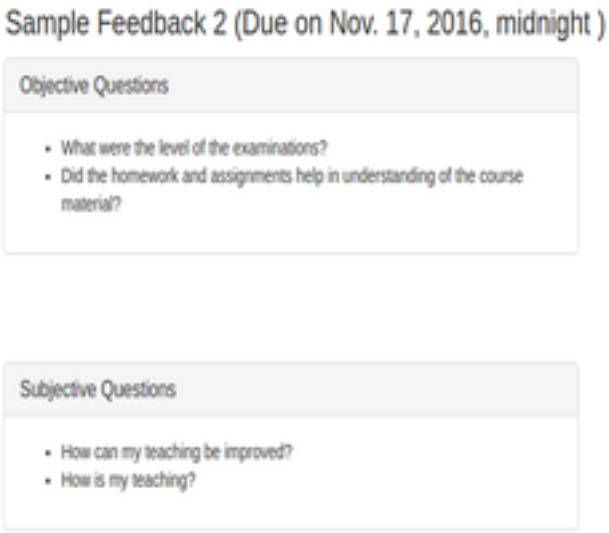
Pending Feedback

Feedback Title	Form Deadline	Questions
Sample Feedback 2	Nov. 17, 2016, midnight	View
Feedback Title	Form Deadline	Questions

Completed Feedback

Feedback Title	Form Deadline	Results
Sample Feedback	Oct. 5, 2016, midnight	View
Feedback Title	Form Deadline	Results





C. Student Interface

1. The authentication to the django server was a main hurdle for us in this task
2. Once the student logs into the app, he sees a calendar view which has the different days which are colur coded on the basis of assignments and deadlines
3. He is also able to see a feedback form
4. A logout feature has been provided

Reflection Essay

This project was the first major project for most of us and certainly on which taught us a lot. Having built websites only in plain-old PHP prior to this, a web framework like Django was a lot easier once we understood how it worked and we got the hang of it. Django has a lot of functionality and very little code is required for it, and this is just what we need in this day and age.

We learnt how to develop user friendly portals, interactive UI and many mnay other features that Django provides.

Regarding Social Login, we hit a brick wall when we tried to do login for Gmail using their API. We spent a lot of time on this before we eventually gave up on Gmail Login and attempted to do facebook login at which we succeeded. In hindsight, this time could have been used more wisely.

Coming to Android, even here we spent almost a whole week trying to do the authentication using oauth and not just plain java. Oauth being more secure was the preferred choice, and although we tried and tried, we finally had to resort to Java to do what we had to do. This I believe was the main reason for not being ablke to complete all the features of the Android Student Interface. Some learnings from the Android Project:-

- Cant use same intent definition in two different functions.
- Get-Post along with Django views makes client-server interaction between them. == is not commonly used in Java. object1.equals(object2) is preferred. Internet requests better be in Background Threads(implemented using Async task), prefer not to use any variable whose value is expected from the thread in our Main thread.
- Shared Preferences:- Usage(helped by Lohith)This project was the first major project for most of us and certainly on which taught us a lot. Having built websites only in plain-old PHP prior to this, a web framework like Django was a lot easier once we understood how it worked and we got the hang of it. Django has a lot of functionality and very little code is required for it, and this is just what we need in this day and age.

  We learnt how to develop user friendly portals, interactive UI and many mnay other features that Django provides.

  Regarding Social Login, we hit a brick wall when we tried to do login for Gmail using their API. We spent a lot of time on this before we eventually gave up on Gmail Login and attempted to do facebook login at which we succeeded. In hindsight, this time could have been used more wisely.

  Coming to Android, even here we spent almost a whole week trying to do the authentication using oauth and not just plain java. Oauth being more secure was the preferred choice, and although we tried and tried, we finally had to resort to Java to do what we had to do. This I believe was the main reason for not being ablke to complete all the features of the Android Student Interface. Some learnings from the Android Project:- (Cant use same intent definition in two different functions. Get-Post along with Django views makes client-server interaction between them. == is not commonly used in Java. object1.equals(object2) is preferred. Internet requests better be in Background Threads(implemented using Async task), prefer not to use any variable whose value is expected from the thread in our Main

thread. Shared Preferences:- Usage(helped by Lohith) -¿ 1. SharedPreferences sharedprefs = this.getApplicationContext(); 2. sharedprefs = getSharedPreferences(Name for reference,0(or 1,2,..)) // (int string,int mode))

All in all, a good project.

1. SharedPreferences sharedprefs = this.getApplicationContext();
2. sharedprefs = getSharedPreferences(Name for reference,0(or 1,2,..)) // (int string,int mode))

All in all, a good project.