

Internship Report (May - June 2018)

e-Acad

Jerri Vaishnavi and Mothe Sravani

June 22, 2018

Supervisor:

Prof. K. Chandrasekaran

Department of CSE, NITK Surathkal



Table of Contents

Table of Contents	1
1. Introduction	3
1.1. Aim and Scope	3
1.2. Stakeholders	3
2. Requirements and Data Model	4
2.1. Functional Requirements	4
2.2. Non-functional Requirements	7
2.3. Requirements Prioritization	8
2.4. Requirements Validation	11
2.5. Data Model	12
3. Development Plan	13
3.1. Process Model	13
3.2. Timeline	13
4. Report of Release 1	15
4.1. Screenshots	15
Login,Signup,Logout for teacher and student	15
Add Course -teacher	16
My courses-teacher	17
Course Page-Teacher	17
Edit Course-teacher	18
Add Class Notes,Assignments-teacher	18
View Submission-teacher	19
Allot Marks-teacher	19
View Course-student	20
My Course-student	21
Course Page-student	21
Submission Link-student	22
View My Submission-student	22
4.2. Review comments	22
5. Report of Release 2	23
5.1. Screenshots	23
New Template	23
Usertype-Teaching Assistant	24

TA course registration	25
Teacher can assign a TA for an assignment	25
Teacher and TA can decide whether students can view marks and feedback of their submission.	25
Discussion Forum for every assignment and class notes	26
Change Password	26
View Profile	27
5.2. Test Plan	27
6. Test cases and results	28
6.1 Test Case for signup page	28
Test Case id : TC001 (Test data for text box)	28
Test Case id: TC002 (Test for buttons)	28
6.2. Test Case for Login page	29
Test Case id: TC003 (Test data for text box and button)	29
6.3. Test Case for Navigation Bar	29
Test Case id: TC004 (Test for links)	29
6.4. Test Case for Uploading Files	29
Test Case id: TC005	29
6.5. Test Case for Alert messages	30
Test Case id: TC006	30
7. How to Use	31
7.1. Teacher	31
7.2. Student	31
7.3. Teaching Assistant	32
8. Report Summary	32

1. Introduction

e-Acad is a website which is developed with an open-source web framework written in python, which follows the model-view-template architecture pattern. Django Packages is a directory of reusable apps, sites and tools, and more for django projects. As Forms are frequently used in the development of this website we used django-crispy-forms package which lets us to easily build, customize and reuse forms without writing template code. This website is implemented with multiple user types.

1.1. Aim and Scope

The main purpose of e-Acad is to implement systematic online course management where we merged all the academic activities. This website is designed to support both teaching and learning where all the class notes, assignments, submissions, permissions, discussions, marks and feedback are managed among the stakeholders. As a whole this website would reduce the involvement of hard copies.

1.2. Stakeholders

- Teacher
- Teaching Assistant
- Student
- Admin

2. Requirements and Data Model

2.1. Functional Requirements

- **FR1:**
Name: SignUp
Description: User can sign up according to his role. User can sign up as a teacher or as a teaching assistant or as a student.
- **FR2:**
Name: Login
Description: User can login to his/her account with his/her unique username and password.
- **FR3:**
Name: Change Password
Description: User can change his/her password.
- **FR4:**
Name: LogOut
Description: User can log out of his account any time.
- **FR5:**
Stakeholder: Teacher
Name: Add Course
Description: Teacher can create a course such that only the user who created the course can manage the course details.
- **FR6:**
Stakeholder: Teacher
Name: Edit Course
Description: Teacher can create, update and delete course details.
- **FR7:**
Stakeholder: Teacher
Name: Add Class Notes
Description: Teacher can add Class Notes(multiple files) for a particular course.
- **FR8:**
Stakeholder: Teacher
Name: Add Assignments
Description: Teacher can add Assignments for a particular course.

- **FR9:**
Stakeholder: Teaching Assistant
Name: Register Course
Description: Teaching Assistants can register for the courses they are interested in during their sign up.
- **FR10:**
Stakeholder: Teaching Assistant
Name: Update Course
Description: Teaching Assistant can update his/her course registration.
- **FR11:**
Stakeholder: Teaching Assistant
Name: View Class Notes
Description: Teaching Assistants can view all the Class Notes uploaded by the teacher if he/she registered for that particular course.
- **FR12:**
Stakeholder: Student
Name: View Course
Description: Students can view and register for the courses offered according to their semester.
- **FR13:**
Stakeholder: Student
Name: My Courses
Description: Students can view the courses that he/she has registered. Students can view the Class Notes, Assignments uploaded by the teacher of that particular course.
- **FR14:**
Stakeholder: Student
Name: Submission link
Description: Students can upload their submission for a particular assignment.
- **FR15:**
Stakeholder: Student
Name: View My Submission
Description: Students can check updates (marks, feedback) of their submissions.

- **FR16:**
Stakeholder: Teacher
Name: View Submission
Description: Teacher can update marks and feedback for all the submissions by the students.
- **FR17:**
Stakeholder: Teacher
Name: Allot Teaching Assistant
Description: Teacher can assign teaching assistants for an assignment to view submissions of all the students so that teaching assistant can update marks and feedback.
- **FR18:**
Stakeholder: Teacher
Name: Permission
Description: Teacher is given a choice that decides whether students can view marks and feedback of their submission.
- **FR19:**
Stakeholder: Teaching Assistant
Name: View Assignments
Description: Teaching Assistants can view assignments only if teacher assign him/her for a particular assignment.
- **FR20:**
Stakeholder: Teaching Assistant
Name: View Submissions
Description: If a Teaching Assistant is assigned for an assignment, he/she can update marks and feedback for submissions of that assignment.
- **FR21:**
Stakeholder: Teaching Assistant
Name: Permission
Description: Teaching Assistant is given a choice that decides whether students can view marks and feedback of their submission.
- **FR22:**
Stakeholder: Teacher
Name: Delete Document
Description: Teacher can delete Class Notes and Assignments that he/she had uploaded.

- **FR23:**
Name: View Profile
Description: Teachers, Teaching Assistants and Students can view and update their profiles.
- **FR24:**
Name: Discussion
Description: Teachers, Teaching Assistants and Students can participate in discussions for a particular class notes or an assignment.
- **FR25:**
Name: Attendance
Description: Teachers list out the attendance of each student, and students can know their attendance percentage in particular semester.
- **FR26:**
Name: My Private Files
Description: Teachers, Teaching Assistants and Students can upload their private files related to the course which are visible only to the respective user.
- **FR27:**
Name: Multiple File Upload.
Description: To upload multiple files at a time.
- **FR28:**
Name: Option to add Field as per the user
Description: For example, while student is submitting a file he/she has to include roll number or registration number etc., depending on the Teacher.

2.2. Non-functional Requirements

1. The system must be able to run without interruption and able to handle thousands of concurrent requests. It must be able to effectively recover from crashes and other problems, as it is handling sensitive data. It should be designed in such a way that it can run on distributed systems.
2. The developed system will have to deal with large quantities of data and a large number of users accessing the data at once.
3. The response time of a search operation should be as less as possible. The value of the limit is not decided yet.

4. Security requirements are important factors in this system as classified data will be stored in the database. User validation will be done during login to insure that the user is valid and that the user only has access to his or her permission data. The system should be designed to make it impossible for anybody to logon without a valid username and password.

This software will,

- Authenticate each user, who logs in:
- When the user performs any action, authorize him/her to perform the activities allowed for the user and displays an error message if found to be unauthorized.
- The system should be easy to use.
- The system should be available 24 hours.
- The system should respond at the time.
- The System should provide specific information to specific user.
- The system should not fail.
- Right information is available to right student and teacher at right time.

2.3. Requirements Prioritization

STAKEHOLDER: ALL

S.No.	NAME	DESCRIPTION	PRIORITY
1	SignUp	User can sign up according to his role. User can sign up as a teacher or as a teaching assistant or as a student.	High
2	Login	User can login to his/her account with his/her unique username and password.	High
3	Change Password	User can change his/her password.	medium
4	LogOut	User can log out of his account any time.	High
5	View Profile	Teachers, Teaching Assistants and Students can view and update their profiles.	Low

STAKEHOLDER: TEACHER

S.No.	NAME	DESCRIPTION	PRIORITY
7	Edit Course	Teacher can create, update and delete course details.	Low
8	Add Class Notes	Teacher can add Class Notes(multiple files) for a particular course.	High
9	Add Assignments	Teacher can add Assignments for a particular course.	High
10	View Submission	Teacher can update marks and feedback for all the submissions by the students.	High
11	Allot Teaching Assistant	Teacher can assign teaching assistants for an assignment to view submissions of all the students so that teaching assistant can update marks and feedback.	High
12	Permission	Teacher is given a choice that decides whether students can view marks and feedback of their submission.	Medium
13	Delete Document	Teacher can delete Class Notes and Assignments that he/she had uploaded.	Medium
14	Discussion	Teacher can participate in discussions for a particular class notes or an assignment.	Medium

STAKEHOLDER:TEACHING ASSISTANT

S.No.	NAME	DESCRIPTION	PRIORITY
16	Update Course	Teaching Assistant can update his/her course registration.	High

17	View Class Notes	Teaching Assistants can view all the Class Notes uploaded by the teacher if he/she registered for that particular course.	High
18	View Assignments	Teaching Assistants can view assignments only if teacher assign him/her for a particular assignment.	High
19	View Submissions	If a Teaching Assistant is assigned for an assignment, he/she can update marks and feedback for submissions of that assignment.	High
20	Permission	Teaching Assistant is given a choice that decides whether students can view marks and feedback of their submission.	Medium
21	Discussion	Teaching Assistants can participate in discussions for a particular class notes or an assignment.	Medium

STAKEHOLDER:STUDENT

S.No.	NAME	DESCRIPTION	PRIORITY
23	My Courses	Students can view the courses that he/she has registered.Students can view the Class Notes, Assignments uploaded by the teacher of that particular course.	High
24	Submission link	Students can upload their submission for a particular assignment.	High
25	View My Submission	Students can check updates (marks, feedback) of their submissions.	High
26	Discussion	Students can participate in discussions for a particular class notes or an assignment.	Medium

2.4. Requirements Validation

There are 28 functional requirements to be implemented as mentioned in Functional Requirements. But out of those only 25 Functional requirements are validated from the customer. Those are From FR01 to FR25 functional requirements.

The functional requirements which are “INVALID”:

- **FR25:**

Name: Attendance

Reason: This is out of the Scope

- **FR26:**

Name: My Private Files

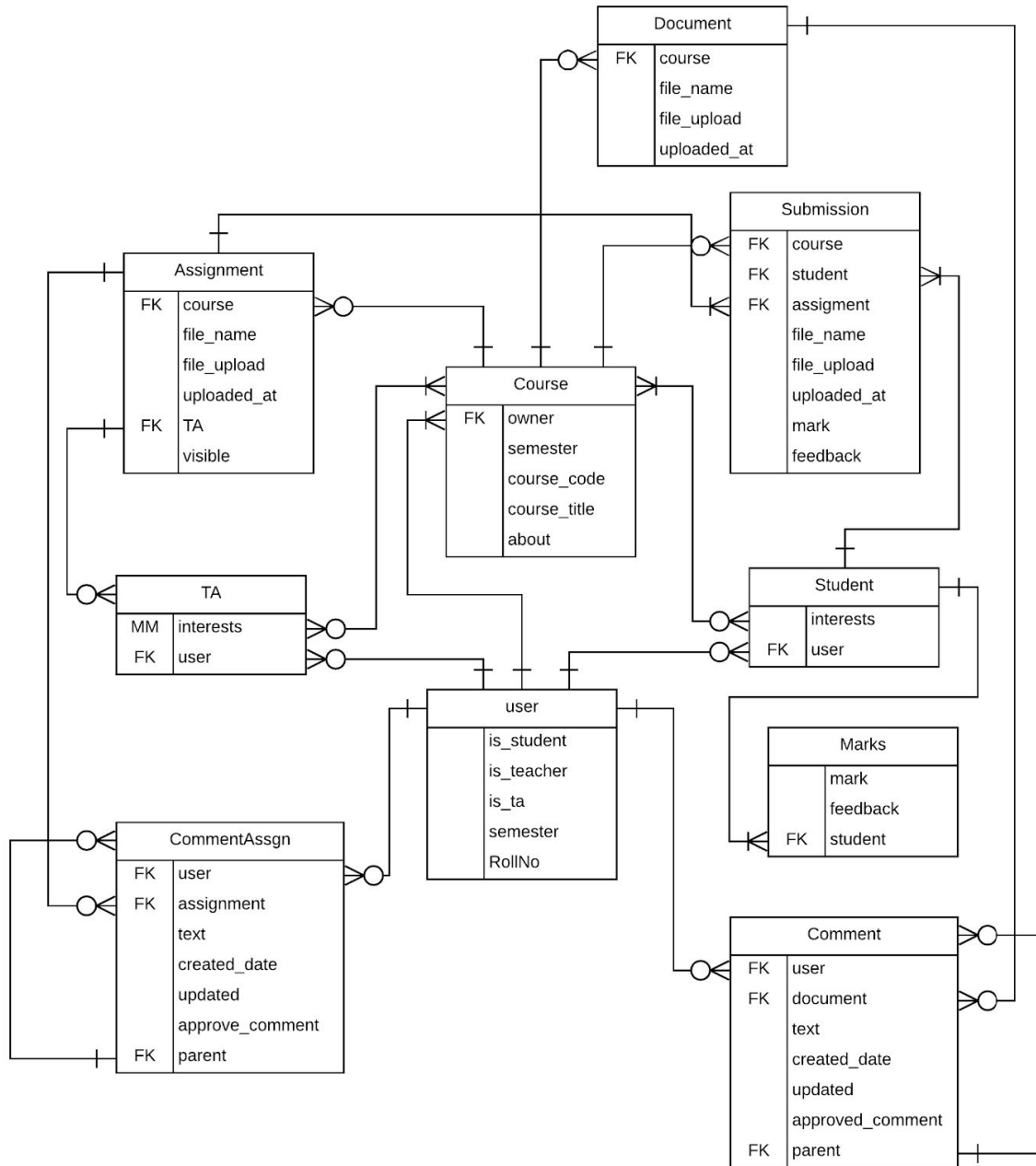
- **FR27:**

Name: Multiple File Upload.

- **FR28:**

Name: Option to add Field as per the user

2.5. Data Model



3. Development Plan

3.1. Process Model

The process model we have chosen here is Agile Methodology.

Reasons:

- To make software development faster and, at the same time have higher customer satisfaction.
- As requirements of the customer are not fixed and to organize the activities properly.
To know review of the customer on each build, so that changes can be made easily.

Here the project is broken up into several 'iterations'. The team members decided the basic core features that are required in the product and decide which of these features can be developed in the first iteration . Any remaining features that cannot be delivered in the first iteration will be taken up in the next iteration, based on the priority. At the end of the each iteration we delivered a working product and depending on the review and feedback on that iteration we switched to other iteration. Everyday discussions were made to analyse the requirements and activities made on that day.

3.2. Timeline

S.no	Work	Start date	End date	Status
1	Requirement analysis,to list out the activities and task. to find out the stakeholders and mapping the stakeholders with the activities	15/05/2018	16/05/2018	completed
2	To list out the valid functional requirements	16/05/2018	17/05/2018	completed
3	To work with the django registration package	17/05/2018	18/05/2018	failed
4	Working with the models, view and sample homepage using django. Brief outline of a data model for the web application	18/05/2018	19/052018	Completed

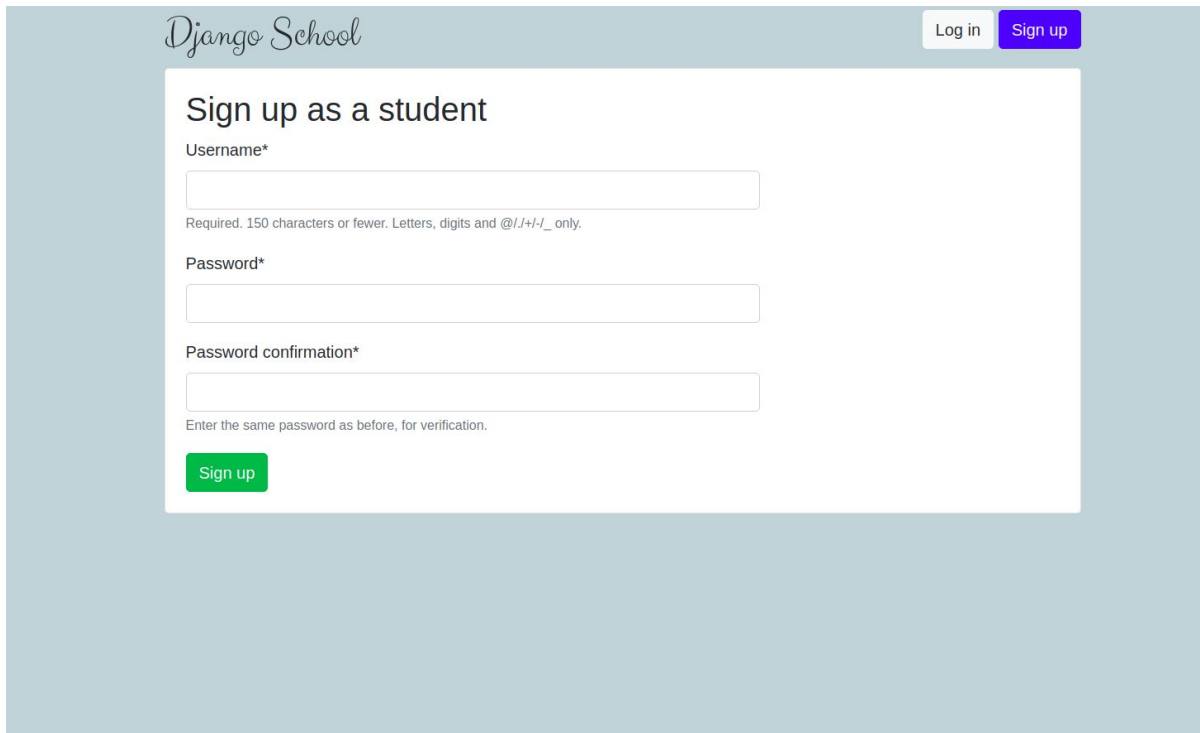
S.no	Work	Start date	End date	Status
5	How to use a template, template engine of django, django template language, template system basics.	18/05/2018	21/05/2018	completed
6	How to use django built-in login system	21/05/2018	22/05/2018	completed
7	Install django school, adding views under teacher and student such that when teacher create a form it should be visible to student	22/05/2018	24/05/2018	completed
8	Teacher register and add courses	24/05/2018	25/05/2018	completed
9	Students register and courses which are added by teacher are visible.	25/05/2018	26/05/2018	completed
10	Student register to the courses available	26/05/2018	29/05/2018	completed
11	Teacher uploads class notes and assignments under a course and those are visible to the registered student	29/05/2018	31/05/2018	completed
12	Students submit the files to a particular assignment for a course	01/06/2018	05/06/2018	completed
13	Teachers allot marks and feedback to that submission and make visible to student	05/06/2018	07/06/2018	completed
14	Adding a new user i.e., Teaching Assistant	07/06/2018	08/06/2018	completed
15	Adding a discussion view for class notes and assignments	08/06/2018	10/06/2018	completed
16	Working on the template	10/06/2018	21/06/2018	completed

4. Report of Release 1

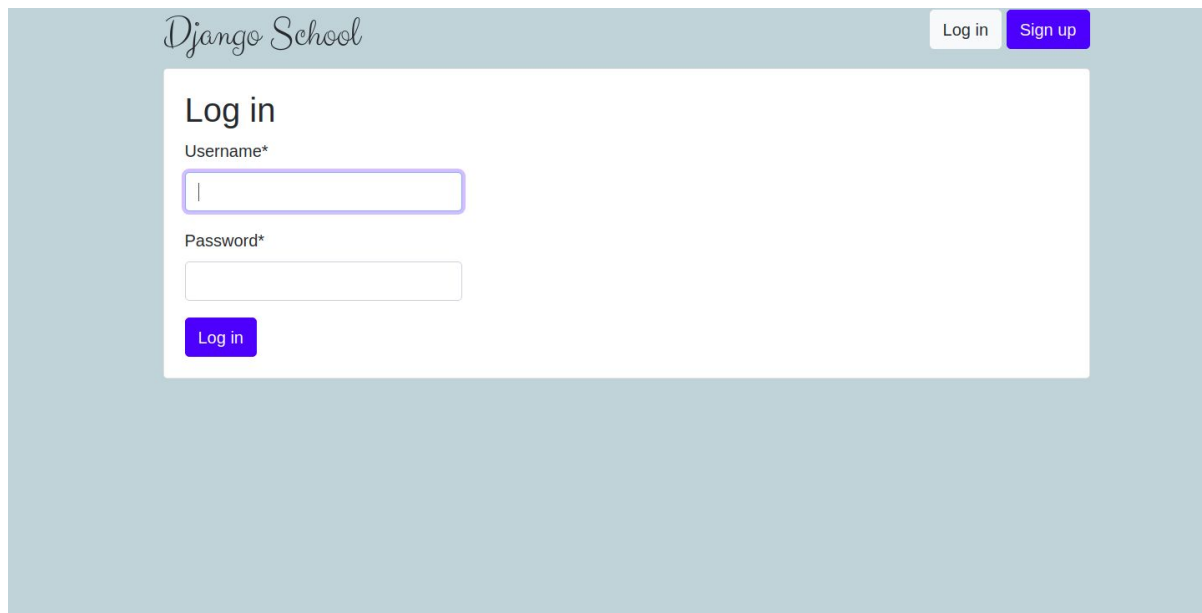
By the end of build-1 the main purpose of this website was implemented. In Build-1 we have two users, teacher and student where teacher can create, update, delete a course, upload class notes, assignments for a course, view submissions and update marks and feedback. Students can register for courses they are interested in and can view all the class notes, assignments of a course. Student can also check updates like marks and feedback of his/her submission.

4.1. Screenshots

1. Login, Signup, Logout for teacher and student



The screenshot displays the 'Django School' website interface. At the top right, there are 'Log in' and 'Sign up' buttons. The main content area features a 'Sign up as a student' form. The form includes three input fields: 'Username*', 'Password*', and 'Password confirmation*'. Below the 'Username*' field, a note states: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.' Below the 'Password confirmation*' field, a note states: 'Enter the same password as before, for verification.' A green 'Sign up' button is located at the bottom of the form.



The screenshot shows the Django School login page. At the top left is the logo "Django School". At the top right are two buttons: "Log in" and "Sign up". The main content area is a white box with the title "Log in". Below the title are two input fields: "Username*" and "Password*", each with a small asterisk indicating it is required. Below the password field is a "Log in" button.

Django School

Log in Sign up

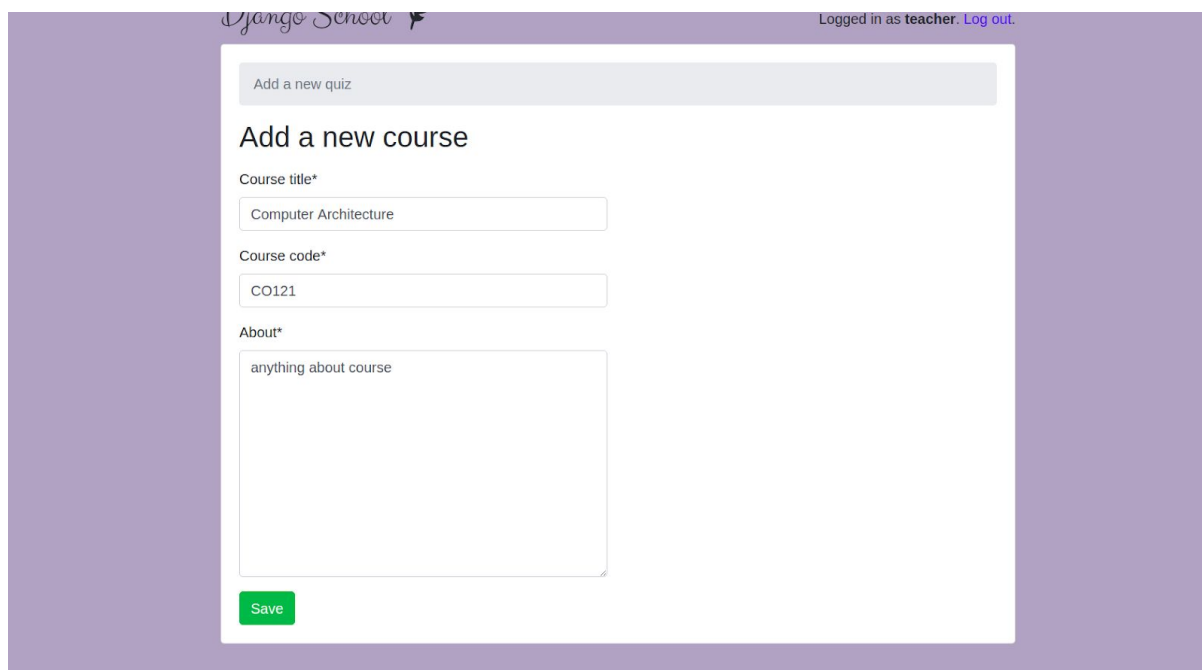
Log in

Username*

Password*

Log in

2. Add Course -teacher



The screenshot shows the Django School "Add a new course" page. At the top left is the logo "Django School". At the top right, it says "Logged in as teacher. Log out." Below the logo is a link "Add a new quiz". The main content area is a white box with the title "Add a new course". Below the title are three input fields: "Course title*" (with the text "Computer Architecture"), "Course code*" (with the text "CO121"), and "About*" (with the text "anything about course"). Below the "About*" field is a "Save" button.

Django School

Logged in as teacher. Log out.

Add a new quiz

Add a new course


Course title*

Course code*

About*

Save

3. My courses-teacher


Django School  Logged in as **teacher**. [Log out.](#)

My Courses

My Courses

Code-Title
CO123-OperatingSystems
CO124-DataCommunications
CO127-Environment Sciences
CO110-Object_Oriented_Programming
CO122-Data Structures
CO121-Computer Architecture

4. Course Page-Teacher

Django School  Logged in as **teacher**. [Log out.](#)

My Courses / DataCommunications

anything about course

[Add Class_Notes](#) [Add Assignments](#) [edit_course_information](#)

Class Notes

Class Notes	Uploaded at	Delete
class_note1	(Uploaded at: June 19, 2018, 8:58 a.m.)	Delete
class_note2	(Uploaded at: June 19, 2018, 8:59 a.m.)	Delete
class_note3	(Uploaded at: June 19, 2018, 8:59 a.m.)	Delete

Assignments

Assignment	Uploaded at	View Submission	Delete
first_file	(Uploaded at: June 8, 2018, 10:55 a.m.)	View Submissions	Delete
gogo	(Uploaded at: June 8, 2018, 11:25 a.m.)	View Submissions	Delete
assignment3	(Uploaded at: June 19, 2018, 8:59 a.m.)	View Submissions	Delete


5. Edit Course-teacher

The screenshot shows a web interface for editing a course. At the top, there is a breadcrumb trail: [My Courses](#) / [DataCommunications](#). Below this, the title 'DataCommunications' is displayed. The form contains three fields: 'Course title*' with the value 'DataCommunications', 'Course code*' with the value 'CO124', and 'About*' with the value 'anything about course'. At the bottom of the form, there are three buttons: 'Save changes' (green), 'Nevermind' (grey), and 'Delete' (red).

6. Add Class Notes,Assignments-teacher

The screenshot shows a web interface for adding class notes or assignments. At the top, there is a breadcrumb trail: [My Courses](#) / [DataCommunications](#) / [Add a new file](#). Below this, the title 'Add File' is displayed. The form contains two fields: 'File name*' with the value 'class_note4' and 'File upload*' with the value 'ticket.pdf'. Below the 'File upload*' field, there is a 'Choose file' button and an 'Upload' button (green). At the bottom of the form, there is a link: [Return to Course_page](#). In the top right corner, there is a message: 'Logged in as teacher. [Log out.](#)'

7. View Submission-teacher

Django School 

Logged in as **teacher**. [Log out.](#)

[My Courses](#) / [DataCommunications](#) / Submissions


Submissions	Student Name	Uploaded at	Marks
submission1	vaish6799	June 19, 2018, 9:08 a.m.	<button>Allot Marks</button>
submission1	student	June 19, 2018, 9:12 a.m.	<button>Allot Marks</button>
submission1	student1	June 19, 2018, 9:13 a.m.	<button>Allot Marks</button>
submission1	student2	June 19, 2018, 9:14 a.m.	<button>Allot Marks</button>
submission1	student3	June 19, 2018, 9:15 a.m.	<button>Allot Marks</button>

8. Allot Marks-teacher

[My Courses](#) / [DataCommunications](#) / [Submissions](#) / student

Submissions	Student Name	Uploaded at	Marks
submission1	student	June 19, 2018, 9:12 a.m.	<div>Mark* <input type="text" value="15.0"/></div> <div>Feedback* <input type="text" value="make changes and submit again"/></div> <div><button>Save changes</button></div>

9. View Course-student

Django School 

Logged in as **vaish6799**. [Log out.](#)

Courses Offered

Courses Offered

Code-Title

Interests*

☒ OperatingSystems

☒ DataCommunications

☒ Environment Sciences

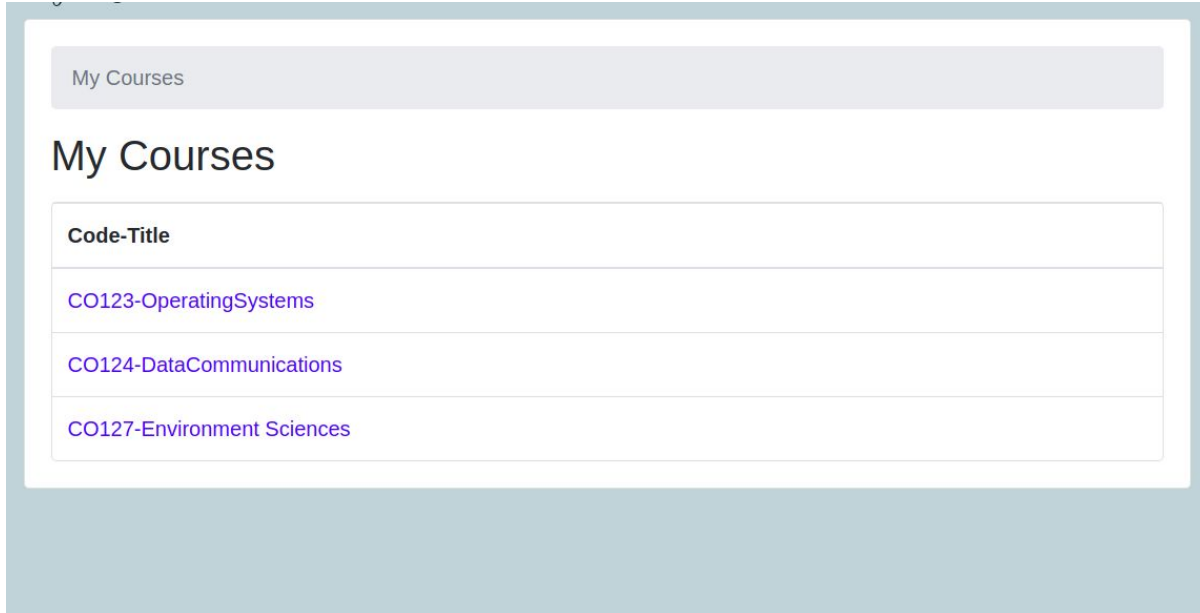
☐ Object_Oriented_Programming

☐ Data Structures

☐ Computer Architecture

Save

10. My Course-student

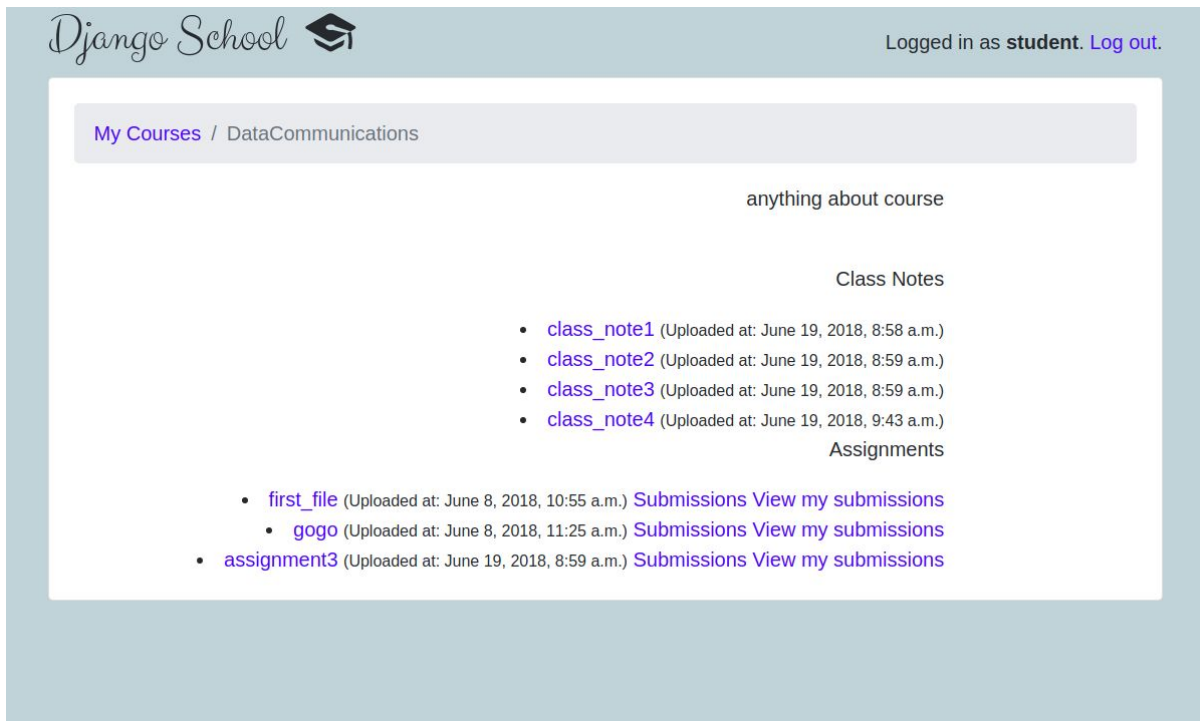



My Courses

My Courses

Code-Title
CO123-OperatingSystems
CO124-DataCommunications
CO127-Environment Sciences

11. Course Page-student



Django School 

Logged in as **student**. [Log out](#).

[My Courses](#) / [DataCommunications](#)

anything about course


Class Notes

- [class_note1](#) (Uploaded at: June 19, 2018, 8:58 a.m.)
- [class_note2](#) (Uploaded at: June 19, 2018, 8:59 a.m.)
- [class_note3](#) (Uploaded at: June 19, 2018, 8:59 a.m.)
- [class_note4](#) (Uploaded at: June 19, 2018, 9:43 a.m.)

Assignments

- [first_file](#) (Uploaded at: June 8, 2018, 10:55 a.m.) [Submissions View my submissions](#)
- [gogo](#) (Uploaded at: June 8, 2018, 11:25 a.m.) [Submissions View my submissions](#)
- [assignment3](#) (Uploaded at: June 19, 2018, 8:59 a.m.) [Submissions View my submissions](#)

12. Submission Link-student

Django School  Logged in as **vaish6799**. [Log out.](#)

Add File


File name*

File upload*

[Choose file](#) Untitled document.pdf

[Upload](#)

13. View My Submission-student

Django School  Logged in as **student**. [Log out.](#)

[My Courses](#) / [DataCommunications](#) / My Submission

Submissions	Student Name	Uploaded at	Marks	FeedBack
submission1	student	June 19, 2018, 9:12 a.m.	15.0	make changes and submit again

4.2. Review comments

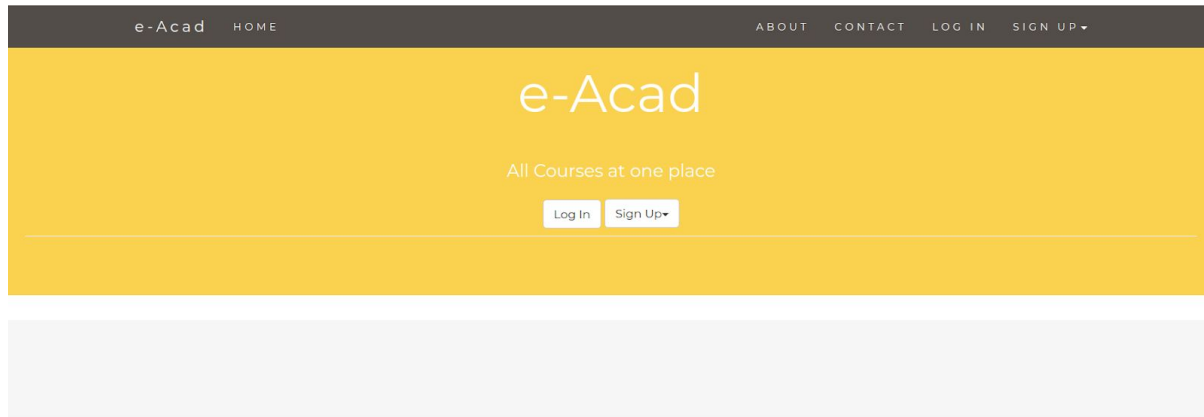
- To add a user of type teaching assistant(TA).
- TA should be able to register for courses and can update his interests.
- TA can view all the class notes uploaded by teacher of a course.
- Teacher should be able to assign teaching assistants for an assignment, so that teaching assistants can view submissions and update marks,feedback for students.
- Teacher and TA should be able to decide whether students can view marks and feedback of their submission.
- A Discussion forum for every class note and assignment uploaded,where students,teachers and teaching assistants can discuss about the files uploaded and submission details.

5. Report of Release 2

5.1. Screenshots

Newly added functionalities:

1. New Template



ABOUT E-ACAD

E-ACAD is an online interactive platform for course management among faculty, teaching assistants and students. The main objective of this platform is to create a personalised learning environment by bringing all the classnotes, assignments and submissions together.



e-Acad
Home
Logged in as teacher

Add ClassNotes
Add Assignments
Edit Course Information

CLASS NOTES

Class Notes	Uploaded at	Discussion	Delete
file1	(Uploaded at: June 20, 2018, 7:53 p.m.)	Discussion	
class_note1	(Uploaded at: June 20, 2018, 8:24 p.m.)	Discussion	

ASSIGNMENTS

Assignment	Uploaded at	Discussion	View Submission	Visible	Allot TA	Delete
assignment	(Uploaded at: June 20, 2018, 8:29 p.m.)	Discussion	View Submissions	Permission	Allot TA	

2. Usertype-Teaching Assistant

e-Acad
Home
Logged in as TA

My Courses

MY COURSES

Update Courses

Semester	Course code	Course Title
1	CO123	OperatingSystems
1	CO125	software_engineering
2	CO127	Object_Oriented_Programming
2	CO125	DataCommunications

3. TA course registration

The screenshot shows a web application interface for a TA user. The header bar is dark grey with 'e-Acad' and 'Home' on the left, and 'Logged in as TA' on the right. The main content area is light grey and titled 'UPDATE YOUR INTERESTS'. Below the title, there is a section labeled 'Interests*'. It contains a list of interests with checkboxes: 'OperatingSystems' (checked), 'software_engineering' (checked), 'Object_Oriented_Programming' (unchecked), 'DataCommunications' (unchecked), and 'Data Structures' (unchecked). At the bottom of this section is a yellow 'Save changes' button. A small yellow arrow points upwards at the bottom center of the page.

4. Teacher can assign a TA for an assignment

The screenshot shows a web application interface for a teacher user. The header bar is dark grey with 'e-Acad' and 'Home' on the left, and 'Logged in as teacher' on the right. The main content area is light grey and titled 'ALLOT TEACHING ASSISTANT'. Above the title is a breadcrumb trail: 'My Courses / OperatingSystems / Allot Teaching assistant'. Below the title, there is a section labeled 'Ta*'. It contains a list of teaching assistants with checkboxes: 'TA' (unchecked), 'ta1' (unchecked), 'ta2' (unchecked), and 'ta3' (unchecked). At the bottom of this section is a yellow 'Save' button.

5. Teacher and TA can decide whether students can view marks and feedback of their submission.

The screenshot shows a web application interface for a teacher or TA user. The header bar is dark grey with 'e-Acad' and 'Home' on the left, and 'Logged in as teacher' on the right. The main content area is light grey and titled 'Marks visible to student?'. Above the title is a breadcrumb trail: 'My Courses / OperatingSystems / Marks visible to student?'. Below the title, there is a section labeled 'Marks visible to student?'. It contains a checkbox labeled 'Visible' which is checked. At the bottom of this section is a yellow 'Save changes' button.

6. Discussion Forum for every assignment and class notes

The screenshot shows the 'Discussion on class notes file1' page in the e-Acad application. The header bar includes 'e-Acad Home' and 'Logged in as Vaishnavi'. The main content area features a title 'Discussion on class notes file1' and a comment input field with an 'Add Comment' button. Below this, a list of comments is displayed, each with a user profile icon, name, timestamp, and content. The comments are as follows:

- teacher teacher** (June 20, 2018, 7:55 p.m.): what is mean by arts
- teacher teacher** (June 20, 2018, 7:55 p.m.): i am adding description
- Teaching Assistant** (June 20, 2018, 9:19 p.m.): here it is!
- vaishnavi Jerri** (June 20, 2018, 9:25 p.m.): thank you :->
- Teaching Assistant** (June 20, 2018, 9:20 p.m.): ask your doubts here
- vaishnavi Jerri** (June 20, 2018, 9:22 p.m.): what is the main purpose of this class note?

Each comment has a 'reply' button. The 'vaishnavi Jerri' comment also includes icons for deleting and editing the post.

7. Change Password

The screenshot shows the 'Change Password' page in the e-Acad application. The header bar includes 'e-Acad Home' and 'Logged in as Vaishnavi'. The main content area features a breadcrumb trail 'HOME / Change Password' and three input fields for password change:

- Old password***: A text input field.
- New password***: A text input field.
- New password confirmation***: A text input field.

Below the input fields is a yellow 'Save changes' button.

8. View Profile

The screenshot displays the 'View Profile' page of the e-Acad application. The page has a dark header with 'e-Acad Home' on the left and 'Logged in as Vaishnavi' on the right. Below the header, a breadcrumb trail shows 'HOME / My Profile'. The main content area contains a form with the following fields:

- Username***: A text input field containing 'Vaishnavi'. Below it, a small note reads: 'Required: 150 characters or fewer. Letters, digits and @/./-/_. only.'
- Semester***: A text input field containing '1'.
- First name**: A text input field containing 'vaishnavi'.
- Last name**: A text input field containing 'Jerri'.

At the bottom of the form is a yellow button labeled 'Save changes'.

5.2. Test Plan

The application requirements are implemented and the important test cases will be tested with suitable test cases and the results are presented in the next section.

6. Test cases and results

6.1 Test Case for signup page

Test Case id : TC001 (Test data for text box)

Sr. No:	Input/action	Expected result	Actual result	Remark
1.	Enter invalid username	Error message “enter valid username ”	Data is accepted without error	pass
2.	Enter invalid email address	Error message “enter valid email address ”	Error message “enter valid email address ”	pass
3.	Enter two different data in password and password confirmation fields	Should not accept the data	Error message “the two password fields did not match ”	pass
4.	Enter username which already exists	Error message “user already exists ”	Error message “user already exists ”	pass

Test Case id: TC002 (Test for buttons)

Sr. No:	Input/Action	Expected result	Actual result	remark
1.	Click on signup button after filling all fields in form	Displays respected homepage depending on the user	Displays respected homepage depending on the user	pass
2.	Click on signup button without filling any one of the field in form	Error message “field is required”	Error message “field is required”	pass

6.2. Test Case for Login page

Test Case id: TC003 (Test data for text box and button)

Sr. No:	Input/action	Expected result	Actual Result	Remarks
1.	Enter correct username and password	Displays respected homepage depending on the user	Displays respected homepage depending on the user	pass
2.	Enter invalid username password	Error message "enter correct username and password"	Error message "enter correct username and password"	Pass
3.	Click on the button after filling the fields	Opens the homepage	Opens the homepage	pass

6.3. Test Case for Navigation Bar

Test Case id: TC004 (Test for links)

Sr. no:	Input/action	Expected result	Actual result	Remarks
1.	Click on link in the navigation bar	It has to return to the correct page	returns to the correct page	pass
2.	moving from one page to another	Navigation bar has to present in each and every page	Navigation bar is present in each and every page	pass

6.4. Test Case for Uploading Files

Test Case id: TC005

Sr. No:	Input/action	Expected result	Actual result	Remark
1.	Click on upload button	File has to be saved in database	Data is saved	pass
2.	Click on upload button	Files has to be saved in particular tables as mentioned in the models	Data is saved at right destination	pass

6.5. Test Case for Alert messages

Test Case id: TC006

Sr. No:	Input/action	Expected result	Actual result	Remark
1.	Click on marks upload button , with marks greater than maximum marks	Error message “ Marks are greater than maximum marks ”	Error message “ Marks are greater than maximum marks ”	pass
2.	Click on delete button	Error message “ delete confirmation ”	Error message “ delete confirmation ”	pass
3.	Click on edit button	Error message “ information updated successfully ”	Error message “ information updated successfully ”	pass


7. How to Use

7.1. Teacher

- Teacher can sign up and create his own Account.
- Teacher can create a course that he wants to offer to students.
- He can mention to which semester the course belong to.
- Teacher can even edit or delete the details of the course that he has created.
- Teacher can add class notes for a particular course which can be accessed by teaching assistant and student.
- A Discussion Forum can be created for every class notes uploaded where teachers ,teaching assistants and students be able to discuss about the class notes.
- Teacher can add assignments for a particular course where every assignment has the following functionalities:
 - Teacher can view all the submissions made by students for that assignment.
 - Teacher can update marks and feedback of all the submissions made by students.
 - Updated marks and feedback would be visible to students only if teacher updates the permission to visible.
 - A Discussion Forum can be created for every assignment uploaded where teachers ,teaching assistants and students be able to discuss about the class notes.
 - One or more teaching assistants can be allotted for an assignment so that teaching assistants would be able to access the assignment and can have the same functionalities as teacher.
 - Teacher can delete an assignment. After deletion no user could be able to access the assignment.

7.2. Student

- Student can sign up and create his own account.
- Student can see the courses offered according to his semester.
- Students can register for the courses and could be able to view the courses he/she has registered.
- Students can view all the class notes and assignments uploaded by teacher of particular course.
- Students can submit their submissions through a link provided for particular assignment.

- 
- Students can view their own submissions and can check their updated marks and feedback.
 - A Discussion Forum can be created for every class notes,assignments uploaded where teachers ,teaching assistants and students be able to discuss.

7.3. Teaching Assistant

- TA can sign up and create his own account.
- TA can view all the courses and register for courses he/she interested in.
- TA can update his course registration.
- TA can access all the class notes uploaded by teacher of a particular course.
- TA can view an assignment uploaded by teacher only if teacher allot him for that assignment.
- If a TA is assigned then he has the same functionalities as that of teacher for that assignment.
- A Discussion Forum can be created for every class notes,assignments uploaded where teachers ,teaching assistants and students be able to discuss.

8. Report Summary

This report summarizes the development activities and outcome of e-Acad. The application is complete and can be deployed for the real used of NITK Stakeholder.

