

Title: Automating PhD Admission process

Group: 4

Member Details :

<u>ROLL NO</u>	<u>NAME</u>	<u>ROLE</u>
CB.EN.U4CSE19218	SAI PRASHANTH	Tester & FrontEnd Developer
CB.EN.U4CSE19219	K.AKSHITHA	Business Analyst& Report Developer
CB.EN.U4CSE19238	RAGHUL JC	Software Engineer
CB.EN.U4CSE19261	VISHAL AS	BackEnd Developer
CB.EN.U4CSE19262	VISHVAJIT S	Project Manager &Software Developer

Abstract:

The new generation of college applicants has practically grown up on the internet. They are used to accessing anything they want instantly, at the click of a button. Automated application software will help your team answer all their queries instantly, send automated updates to them and help them navigate through the whole application process through a clean and modern interface. Our project provides a software solution to automate the PhD admission processes at research organizations / educational institutes. It provides a central place for all admission related data and it serves as a collaboration platform for all relevant stakeholders. The end result from this software is where best and eligible students for pursuing PhD are selected and called in for an interview.

Stakeholders

- Institution members(Faculty)
- Students who have applied(Applicants)

Tables:

- Student(Master table) – basic information, Purpose : To get the personal information from students to Institute.
 - **Primary Key** – Enrollment ID
 - **Foreign Key** – Subject ID (Subject), Student ID (Payment)
- Student – qualification, Purpose : To get educational information from students.
 - **Primary Key** – Enrollment ID
 - **Foreign Key** – Enrollment ID (Basic Information)
- Teachers(Master table),Purpose : Access students details and sort them for admission purpose.
 - **Primary Key** – Teacher ID
- Results, Purpose: Show the status of admission of students.
 - **Primary Key** – Enrollment ID
 - **Foreign Key** – Enrollment ID (Basic & Qualification)
- Subjects, Purpose : To have details about the maximum number of seats available in each subject.
 - **Primary Key** – Subject ID
 - **Foreign Key** – Enrollment ID (Basic information), Teacher ID (Teachers)
- Payment, purpose: Application processing fee.
 - **Primary Key** – Enrollment ID
- Interview, purpose: Organize interview process for selected students.
 - **Primary Key** – Enrollment ID
 - **Foreign key** – Enrollment ID(Results)

Reports:

- **Results**
 - **Purpose:** Display status of students i.e Selected, Rejected, Wait list. Display the allocations and availability of subjects being selected.
- **Software Requirement**
 - **Purpose:** A report from business analyst to developers on what the requirement is.
- **Software Update**
 - **Purpose :** From developers to business analyst / project manager regarding the software updates
- **Testing Report**
 - **Purpose:** Tester's report for developers regarding errors.
- **User feedback**
 - **Purpose:** From users to project manager regarding their feedback on using the software.

Technologies Used:

Frontend-HTML,CSS, JavaScript, Java

Backend-Firebase

Chart preparation-InVision

Assumptions:

- Students might ask for the track of their application process, which we aren't going to provide.
- Students might ask for the review / opinion of college, which we aren't going to provide as we are just clients and not a member of the educational institute.
- iOS version of mobile application is not provided.

- Merely submission of an application form, does not constitute a claim for an offer date for the selection procedure or admission to the program. The form may be rejected due to deficiency in the supporting documents, not following the prescribed procedures or any other valid reason.
- As a student, you can personally contact teachers regarding admission, which is not provided.

➤ **User--Student (Add/Delete/Modify/Update)**

1. View application
2. Upload documents
3. Payment method
4. Check Confirmation of payment
5. Check whether their application is submitted
6. Login using credentials
7. Print their application

➤ **User--Teacher (Add/Delete/Modify/Update)**

1. View their profile.
2. Access to use applications of students.
3. Update status of applications.
4. Login using credentials

➤ **Instructions**—Post important messages from educational institute to user (teachers and students).

Software:

1. Webpage
2. Mobile application(Android)
3. Database

Hardware:

1. Pc with stable internet connection
2. An android phone with stable internet connection.

Steps involved from stakeholder's point of view:

- **Student's point of view:**
 - A student who wants to apply for a college uses this software.
 - Student has to create an account login using his mail id and a password created and there are many modules.
 - The main module is filling out the application. There are two types of requirements. Personal data with id and documents and educational data with proof. All details should be filled correctly and if not validated properly by the user , then you have to pay again to submit the application.
 - After filling out the form, an enrollment id is given. That acts as the unique, primary key. Students use their enrollment id to view their results.
 - When submitting a fee of a particular amount depending upon the institute is asked to be paid. This is considered as application fee. The method for payment is G-pay.

- After payment, application is submitted and student can check whether application is submitted or not. This application can be also printed.
- After that he can't track application whether it's been viewed and selected or not.
- During the day of result he can check his application form and there will be a result whether he is selected or rejected or in waiting list.
- After results are out, there will be a new module where user can see his time of interview which can't be changed or altered. This is shown only if he is selected.
- **Faculty's point of view:**
 - Faculties also have a login method. Using their mail id / college id and password.
 - They have to update their personal details in profile module.
 - Depending on their designation and subject, they are assigned on whether they can view student's application.
 - A special set of faculties – admission committee only have the permission to publish results on whether a student is selected or rejected or in waiting list with consult of faculties. This consultation is done manually and not through the software (Let's say they all are in an office and discussing about it).

EPIC-STORIES-TASKS:

1. EPIC : LOGIN AUTHENTICATION

1.User Stories: As a new user,I need to register by creating username and password so that system can remember me and my data .

Tasks:

- Design Registration page with name, email ID, phone number as input fields.
- Cut SVG icons and images.
- Option to register as a student/faculty.
- If not a new user, provide an option to login page.
- Implement login page HTML/CSS/JAVA
- Use firebase to store data.
- Use data in firebase: email id to send verification link.
Phone no. to send OTP for verification.

2.User Stories: As a student, I need to login to my account with username and password so that the system can authenticate me and I can trust it.

Tasks:

- Design login page using HTML/CSS/JAVA.
- Design a splash screen.
- Use SVG icons and images.
- Login is first module after splash screen.
- After login students are p
- Use firebase to fetch and match the login credentials from register.

3.User Stories: As a user I need an application ID in order to link my application status to my admission application.

Tasks:

- Design a separate text view and button.
- Cut icons and images for button.
- Clicking button will generate unique ID.ID datatype is alphanumerical.
- Implement module/function using HTML/CSS/JavaScript.
- Use firebase to store unique ID to database.
- Size of the unique ID is client's request.
(Depending on number of applications allowed in that institution.)

4.User Stories: As a registered user I want to occasionally reset my password so that I can keep it secure.

Tasks:

- Implement password reset page HTML/CSS/JAVA.
- Add a link in login page to a separate page to change password.
- Entered user information is verified from the database.
- A link will be sent to registered email/phone.
- Use firebase to update data.
- Password can't be same as previous password.

5.User Stories: As a faculty from institution, I need a separate Login to my account with university mail ID.

Tasks:

- Design login page for faculty.
- Use icons and images.
- Login is first module after splash screen.
- Implement login page HTML/CSS/JAVA
- Use firebase to fetch and match the login credentials.

6. User Stories: As a registered user I want to be able to request a new password if I forget the password so that I don't lose access to my data.

Tasks:

- Design forget password page using HTML/CSS/JavaScript.
- If new user, redirected to registration page.
- Use data in firebase: email id to send password reset link. Phone no. to send reset link SMS for verification.
- Use firebase to update previous data with new password.

2. EPIC: FILL ONLINE APPLICATION

1. User Stories: As an applicant, I need to view instructions and check agreement policy so I can proceed with the application process.

Tasks:

- Create a page with all the details like admission procedure, fee structure, eligibility criteria, contact details of the institution.
- Put SVG icons and institution related images.
- Implement this page using HTML/CSS/JAVA
- Back button to go to home page.
- Add necessary links which redirects user to a particular module.

2. User Stories: As an applicant, I need to be able to fill the form with my personal details.

Tasks:

- Design personal details forms page using HTML/CSS/JAVA.

- Cut SVG icons and images.
- Use firebase to store data.
- Back button to go to dashboard
- Progress Bar to show progress

3. User Stories: As an applicant,I need to fill all my educational details and upload required documents.

Tasks:

- Design educational details form using HTML/CSS/JavaScript.
- Cut SVG icons and images.
- Use firebase to store data.
- Users to add research paper id or researcher(user) ID .
- Use upload button to store educational certificates(12th, college degree).
- Display the size limit of the files to be uploaded.

4. User Stories: As an applicant, I need to validate my application.

Tasks:

- Design a validate button.
- Module to check criteria for each input.
- Image type should be same as asked.
- Size of image should be within criteria.
- Datatype entered should be within criteria (name should not contain number, CGPA should not contain character). No special characters to be used.
- All forms should be entered. All field are mandatory to be filled.

5. User Stories: As an applicant,I want to view all the Ph.D. research subjects that are being offered by the institution, so I can choose from it.

Tasks:

- Design a subject requested form.
- Display all the subjects available in the institute for PhD.
- Select subjects in the list offered.
- Warning saying FCFS.
- Display Pre requisite / portions for each subject.

6. User Stories: As an applicant, I want to view and download my application to my local machine.

Tasks:

- Design a download button using HTML/CSS/JAVA.
- Add a view application button which redirects me to enter application ID page.
- On inputting the application ID match it with data in database and fetch the application.
- Display the respective application.

3. EPIC: APPLICATION PAYMENT

1.User Stories: As an applicant,I should be able to pay my application fee so that I can finish my application process.

Tasks:

- Design application payment page.
- Cut SVG icons and images.
- Application fee is set by default,users cannot change it.
- Redirect to payment modes
(GPay,credit/debit/netbanking) option to pay.
- Pay button to pay application fee to college.

2. User Stories: As an institutional body, I should be able to receive all the payments to the merchant bank account so that I can ensure smooth functioning of application process.

Tasks:

- Update details of merchandise by the institution.
- After applicant makes a payment the user information has to be sent to the merchant acquiring bank to process.
- Merchandise bank account details should be institute's bank account details.
- Merchandise amount for each application will be updated.
- Should be able to generate reports on payments made.

3. User Stories: As an applicant, after fee payment I should get confirmation message on the status of my payment along with e-receipt.

Tasks:

- Design status button which provides me payment status (successful/denied).
- If paid, a data will be added in firebase saying payment: Boolean true or false.
- After successful payment redirect to registration page with payment checked in status.
- After fee payment provide an option to generate E-receipt.
- After this submit button will be visible, applicant can finally submit the application. (Till then it will be greyed out).

4. User Stories: As an applicant, if payment is failure, my whole process shouldn't be cleared; it should be saved so that I can retry payment.

Tasks:

- Implement payment page HTML/CSS/JAVA
- Use firebase to store temporary data.
- Provide a retry payment button.
- If paid, a data will be added in firebase saying payment : Boolean true or false.
- After payment redirect to registration page with payment checked.
- After this submit button will be visible.(Till then it will be greyed out).

5. User Stories: As an applicant, if my payment is successful and I can't see the result in software, I should have an option to look for support (HELP BUTTON)

Tasks:

- Design help button HTML/CSS/JAVA.
- Cut SVG icons and images.
- Use firebase to store data.
- If paid, a data will be added in firebase saying payment : Boolean true or false.
- Status button to check payment status.
- Help button will redirect me to chatbot/send phone number of administration to applicants mail ID for further enquiries.

TEACHER'S POV:**4. EPIC: REVIEW APPLICATION**

- 1. User stories:** Granting permission to access applications to teacher.

Tasks:

- Cut SVG icons and images.
- Separate login page for faculties, design page using HTML/CSS/JAVA
- Limited access to teachers.
- Should have permission to view table. Use firebase to store privileges
- Design raise request button to gain access
- Establish communication to acknowledge requests

2. User stories: As a permitted teacher I should be able to see the applications

Tasks:

- Check for permission to access the list
- If yes allow
- If no, can request access. Wait till further notice
- Design view table button to view application list. design page using HTML/CSS/JAVA
- Cut SVG icons and images
- Retrieve data from firebase

3. User stories: As a permitted teacher I can sort the applications based on criterias.

Tasks:

- To sort applicants based on the department they have applied
- Design button to sort
- Cut SVG icons

- Use SQL commands to display necessary data from firebase
- Design button to change ascending/descending

4. **User stories:** As a permitted teacher I should be able to print details : details of students, details of applications of a subject

Tasks:

- Select a particular student and print his details
- Select a subject and print students applied for it
- Design print button
- Retrieve data from firebase
- Cut SVG icons and images
- Design page using HTML/CSS/JAVA

5. **User stories:** As a permitted teacher I should be able to check for validation of application and if any error, I can reject it before selection starts.

Tasks:

- If application lacks details or contains a different datatype, they can stop the application from getting into selection process
- Send warning message to the respective student
- Design text window popup for teachers to fill out the reason for pending approval
- Establish communication between admin and teacher
- Cut SVG icons and images

- Give approval when application is valid

5. EPIC: SELECTION PROCESS

1. User Stories: As a teacher I should view the research the student has done and relate it with his application to subject.

Tasks:

- Checking what type of research and which course or branch the applicant is taking.
- If matched going on to the reviews and publication (trying to check legit or not).
- If matched going on to the reviews and publication (trying to check legit or not).
- Giving the score based on the above details.(Helps in sorting with other applicants)
- Sorting based on scores.
- Cut SVG icons and images.
- Implement page using HTML/CSS/JAVA

2. User Stories: As a teacher I should check if application is meeting the criteria set by institute.

Tasks:

- Institute would have set some minimum criteria.
- Checking the minimum criteria.
- If application lacks details or data is untrue, teachers can reject that application.
- Design text window using html/java.

- Design text window popup for teachers to fill out if they need to get approval for rejection.
- Establish communication
- Cut SVG icons and images

3. **User Stories:** As a teacher I should be able to update results in each application.

Tasks:

- Updating every application with a new attribute- status
- This status would contain whether they are Selected, Rejected or in waiting list.
- If selected, applicant will have a button to be redirected to interview schedule.
- If in waiting list, applicant will get a pop up message asking them to wait until interview is done.

4. **User Stories:** As a teacher I should be able to sort applications based on criteria.

Tasks:

- Sorting application based on various criteria.
- Design button to sort
- Cut SVG icons
- Use important columns as categories to filter
- Use SQL commands to display necessary data from firebase
- Design button to change ascending/descending

5. **User Stories:** As a teacher I should be able to add applications to waiting list

Tasks:

- Teachers can select students and add them to waiting list where their approval is decided later on.
- Check for permission

- Design add to waiting list button
- Cut SVG icons and images
- Store data in firebase
- Send message to student

6. User Stories: As a teacher I should be able to reject applications

Tasks:

- If application lacks details or is found to be suspicious, teachers can use it to send warning
- Design text window
- Send warning message to the respective student
- Design text window popup for teachers to fill out if they need to get approval for rejection
- Establish communication
- Cut SVG icons and images
-

6. EPIC: INTERVIEW ALLOCATION

1. **User Stories:** As a selected student, I can request to change date and time of Interview with valid reason.

Tasks:

- Design Interview process page - Dashboard
- Cut SVG icons and images.
- Implement page HTML/CSS/JAVA
- Use firebase to store the request date and time of student.
- Student can select date and time and select Request button

- Student should be able to see a request button which will make them request for a new time.

2. **User Stories:** As a teacher, I should schedule interview based on the application selection.

Tasks:

- Design Interview Allocation page – dashboard.
- Cut SVG icons and images.
- Implement Interview Allocation page using

HTML/CSS/JAVA

- Use firebase to store and retrieve data.
- Based on students' application after sorting, interview will start with most eligible student.
- Display Date and time of Interview

3. **User Stories:** As a selected student, I can view date and time of my Interview

Tasks:

- Design Interview Allocation page – dashboard.
- Cut SVG icons and images.
- Implement Interview Allocation page using

HTML/CSS/JAVA

- Use firebase to store and retrieve data.
- Student can view allocated date and time of their

Interview

- Back button to go to login page

4. **User Stories:** As a teacher, I can update and edit date and time of selected student's Interview

Tasks:

- Design Interview Allocation page – dashboard.
- Cut SVG icons and images.
- Implement login page HTML/CSS/JAVA
- Use firebase to store, update and retrieve data.
- Teachers can allocate the date and time of their student's Interview
- Back button to go to login page

5. **User Stories:** As a student, I can reject the Interview with valid reason

Tasks:

- Design a button for reject the Interview.
- Implement login page HTML/CSS/JAVA
- Student should give a valid reason to reject Interview.
- Pop up window will display for confirmation
- Use firebase to remove that student data.
- Student's account will delete automatically.

EPIC-USER STORIES CHART

