IRIS Web Team Recruitments 2024

Submission Deadline

14th Jan 2024
11:59 PM IST

General Instructions

- **Applicants are to submit any one of the given tasks.**

- To submit your application, create a **private** Github repository with the format of the title as *IRIS_Rec23_Roll-No_Framework-Used*. For example, *IRIS_Rec23_201CS132_Ruby-On-Rails*. Add the following GitHub handles: mittal-parth, hrushikeshj, BenzeneAlcohol, VedantTarale, anirudhprabhakaran3, hgupta12, sanjeevholla26, ndhruv03, Minank-KP, Aditya-1208, Arif-Kalluru as **collaborators** to the repository (any 5).
- Submit the form in IRIS before the deadline. Make sure you submit it, even if it's incomplete/partially completed. Implementing all features is recommended but not necessary. Individual tasks have points too.
- Use the **MVC architecture** (Ruby-on-Rails/Django/MEAN Stack/PHP, etc.) to implement the task.
- Frontend libraries/frameworks like Bootstrap, TailwindCSS, Vue, React, etc., can be used for a more user-friendly experience.
- Feel free to use any online references.
- Feel free to contact any of the PoCs if you need clarification about the task or have any questions.
- When submitting, add a **README.md** which includes the following, though not limited to:
    a. A Demo Video showcasing all features of your project - **Mandatory**
    b. Installation instructions to set up the project from scratch - **Mandatory**
    c. Complete steps to run your project - **Mandatory**
    d. List of implemented features
    e. List of non-implemented/planned features
    f. List of known bugs
    g. References used
    h. Screenshots (if any)

- Good programming practices like modular code, documentation, and following the framework's conventions are recommended and will mean better chances of an interview.

Eligibility

- 1st, 2nd and 3rd year **B. Tech** students (**Batch of 2025, 2026 & 2027**) of any branch are eligible to apply for recruitment to the Web Team.

Task 1: Service Requests

Overview

Often a Student goes from department to department asking for approvals on their request to get a document. What if this entire process could be digitized?

As part of the process of getting any official work done in the college one has to go through an approval flow. The idea of this module is to digitize all sorts of approval flows throughout the college, be it for changing student records or getting transcripts.

For this task, you are to design a web application that allows the admin to make custom request templates having role-based approval flows and the applicant to apply to these templates. You are to also create a working template with a practical use case and approval flow. For example, changing the date of birth with the approval flow being:

Student => Academic Section => MIS Officers

User Characteristics

Student

a. Name
b. Email
c. Department
d. Program Type
e. Student ID
f. Password

Employee

a. Name
b. Email
c. Department
d. Staff ID
e. Password
f. Role

ALL the users must be logged in for ANY activity to happen on this website.

Request Characteristics

Template

a. ID
b. Name
c. Approval Flow(You can implement this however you want)
d. Archived
e. Application Start Time
f. Application End Time

Request

a. ID
b. Template ID
c. Applicant response
d. Status
e. Current Step
f. Participants(i.e., the applicant, and the users who forwarded/approved the request)

These characteristics are not necessarily “attributes”. It is left for the developer to implement these however they see fit. It must be ensured that the functionality remains the same.

Part I: Roles and Request Templates

Roles

1. There are multiple roles an individual can have, for example, Club Convenor,Faculty Advisor and MIS Officers.
2. Each User be it a student or employee is given roles based on which the approval flow is later built

Request Template

1. For each request, there exists a template that defines all the essentials of the request.
2. It has important details like the name of the template, the approval flow which contains the details of what role can apply for this request template, who are the forwarders and approvers and what actions they can take(approve/ reject/ forward etc), and the Application Start and End Time.
3. Additional details such as limits on the number of applications from a single user can also be included.

Part II: Applicant and Forwarder/Approver Interface

Once a request template is created, allow only the applicant role to apply for the request. The request should then move to the next role in the approval flow. Care must be taken to ensure that only a user with the role associated with the current step of the application can take action on the request.

Hints:

1. Check for the user role and compare it with the role of the current step in the approval flow.
2. As a user can have multiple Roles do not allow the user to take action on the request at any stage if the user is the applicant who made the request.

You can come up with alternatives to implement the same.

Part III: Authentication and Authorization

Authentication

1. All users must be logged in to access the web app.
2. You can use any existing service for authentication.

Authorisation

1. Multiple roles exist for example:
    a. **Student**
    b. **Admin -** can create/manage request templates
    c. **MIS Officers -** can create/manage request templates
    d. **Academic Section**

    You can create any roles as you see fit however, there must be one role that has the access to create request templates.

Part IV: Bonus Tasks (Optional)

1. Allow for user-based approval flow. In case of a request being forwarded to a HoD, you would want the request to be accessible only to the HoD of the applicant's department.
2. Allow for the forwarder/approver to comment on the request made by the applicant.
3. Provide an option to send back the request application to the applicant for making changes.
4. Add any other feature that you'd like to. We would love to see your creativity!

Note: The main aim of this task is to test the role-based approval flow. Other aspects can be kept fairly simple.

Resources

1. Authentication -
    a. Rails - https://guides.railsgirls.com/devise
    b. Django - https://docs.djangoproject.com/en/4.0/topics/auth/
    c. Express - https://www.passportjs.org/
2. Authorisation-
    a. Rails - RailsApp
    b. Django - Djangoproject
    c. Express - https://www.npmjs.com/package/express-mailer

2: Course RegistrTaskation

Overview

The academic year is approaching, and the university needs an efficient system to handle course registrations seamlessly. The challenge is to develop a platform that enables students to register for courses they desire while ensuring ease of use for both students and administrators.

Long ago, course registration used to happen manually in NITK. People used to stand in queues just to register for the courses. But with digitization, things have improved. You are free to come up with everything other than the basic requirements on your own.

User Characteristics

Student

g. Name
h. Email
i. Department
j. Program Type
k. Student ID
l. Password

ALL the users must be logged in for ANY activity to happen on this website.

Faculty

g. Name
h. Email
i. Department
j. Staff ID
k. Password

These characteristics are not necessarily “attributes”. These have been discussed in detail below.

Name, Email

Primary identifier for any user.

Department

Department to which the user belongs.

Program Type

The program that the student is studying in - B.Tech/M.Tech/PhD/B.Arch so on. A faculty can teach multiple program types, they are not restricted to a single program type unlike a student.

Password

Self Explanatory

Course Characteristics

Course

a. Course Code
b. Course Title
c. Instructor
d. Schedule
e. Credits

Course Code, Course Title

Primary identifier for any course.

Instructor

The faculty who is taking the course.

Schedule

The actual schedule of the course.

Note: The attributes mentioned here need not be a part of any table. You are free to create your own tables depending on the need and requirements, and are encouraged to do so! These are just the basic requirements that will ease your process of development.

Part I: Course Registration

Student Registration

1. Students must log in to access the course registration system.
2. Upon login, students can view available courses based on their program/major.
3. Students can search for courses by code, title, or department.
4. They can select desired courses and add them to their course cart.

Course Availability & Management

1. Faculty/Admin should be able to add new courses to the system (Faculty can add courses of his/her department only) .
2. Set available slots for each course.
3. Specify prerequisites and additional course details (Basically the description)

Enrollment & Validation

1. Students can submit their course selections.
2. The system checks if students meet prerequisites for selected courses.
4. Next, the system forwards this request to the course instructor.
5. The instructor gets the list of students who have shown interest, and can choose whether to enroll them or drop them.

Part II: Functions of each role

The functions are so on:

1. Admin Functions
    a. Admin can view all courses, student enrollments, and available slots.
    b. Add/Remove courses.
    c. Manage faculty details and assignments.

    d. View enrollment statistics and generate reports.
1. Faculty functions:
    a. Instructors can access courses they're teaching.
    b. View enrolled students.
    c. Modify course details (if allowed by admin).
2. Student functions:
    a. Students can view all courses, and choose to get enrolled.
    b. Check course schedules.
    c. Drop courses within the specified deadline.

Part III: Bonus Tasks (Optional)

Further assessments to the courses

- Once the course has been approved, the student should now be shown a grades section where they can see the marks and grades on different tests.
- This also means the faculty should get to add different types of exams, marks, grading etc, which makes this a little lengthy. However, having a set of predefined mid-sem and end-sem grades can also be done.

Notifications & Reminders

- Send automated emails or notifications for enrollment deadlines, dropped courses, or any updates.

Course Feedback

- Allow students to submit feedback or ratings for completed courses.

Tasks in the Bonus Section can be attempted in any order. It is not compulsory to complete all of the bonus tasks that are mentioned, but do implement as many as possible for better chances of an interview.

Course registration is one of the most important uses of IRIS. Since all of you have gone through course registration before, you can try to mimic as many features as possible, in your own way. They need not be an exact copy, but something with a small handy feature that you think should get added. 🙂

Resources

1. Hybrid Schemas - https://www.stratoscale.com/blog/dbaas/hybrid-databases-combining-relational-nosql/
2. Authentication -
    a. Rails - https://guides.railsgirls.com/devise
    b. Django - https://docs.djangoproject.com/en/4.0/topics/auth/
    c. Express - https://www.passportjs.org/

3. Authorization-

a. Rails - RailsApp
b. Django - Djangoproject
c. Express - https://www.npmjs.com/package/express-mailer

Task 3: Multi-Page About Us

Overview

Over the years, IRIS has evolved into a much larger project, spanning multiple modules, sister projects and initiatives. Owing to this, the current About Us page calls for improvements.

Your task is to improve/redesign the About Us Page, giving it a modern touch but more importantly, making it more descriptive.

You can come up with your own design or build on an existing template.

Part I: The Landing Page

1. Taking inspiration from the existing page, the following sections must stay:
    a. Hero Section

    b. Stats
    c. Modules
    d. Other Projects
    e. Integrations
    f. Team Info
    g. Shoutout to CSD and NITK
    h. Contact Us
2. Other existing sections can be added or omitted at your discretion. New sections, like a Twitter deck, are also welcome.

Part II: Module-wise pages

Currently, it's a single-page website. The idea is to be able to showcase each module with a separate page. This is useful to display the important features of the module and also the people who worked behind it. This can also be a way to showcase the work put in by the team, as most of IRIS work is closed-source.

1. Each module mentioned in the first landing page must link to its own page. (For this task, you can create a page about just one module **completely**. All other modules can follow the same template. But this one page must be done coherently.)
2. Each module page can contain the following:
    a. Name of the module
    b. Why the module was created (problem statement)
    c. Important Features of the module with high-quality screenshots/videos/GIFs (or any other high-quality format like Lottiefiles)
    d. Information about the developers and product managers who worked on that module.
    e. Any other section that you feel is suitable.

Part III: Bonus Tasks (optional)

1. Create a Content Management System (or use a plugin), that allows you to add the content from a UI instead of creating a page for every module. There would be a

template that would exist for each module, and the content can be added/edited directly from this panel without the intervention of a developer.
2. Add animations while making sure that the website page load time doesn't increase significantly.

Important Notes

1. This is a frontend-heavy task and the focus would be on testing the candidate's UI/UX skills.
2. Just a Figma design is NOT accepted.
3. The assets and content must NOT be dummy/placeholder text. Actual screenshots or videos must be taken from IRIS.
4. The design should be responsive and compatible with most screen sizes.
5. Feel free to design as vividly as possible while just making sure that the page load time doesn't increase significantly.
6. **Any** tech stack can be used.

Resources

1. LottieFiles - https://lottiefiles.com/
2. From Idea to Design - https://dev.to/abbeyperini/from-idea-to-design-for-non-designers-m6d
3. Inspirations (these are just places to get started, your exploration is more than welcome)
    a. https://codedamn.com/ -
    b. https://cuvette.tech/
    c. https://push.org/
    d. https://polygon.technology/
    e. https://www.samudai.xyz/
    f. https://www.superfluid.finance/
    g. https://sentry.io/welcome/
    h. https://cosh.nitk.ac.in/
4. React Animation Library - https://www.framer.com/motion/

5. TailwindCSS

Points of Contact

- Muthukumar - +91 89214 72523
- Vedant Anup Tarale - +91 81067 13107