Website Hackathon

Guidelines:-

1. Develop an API-based backend for the website and use it to serve content to frontend(s).
2. Please provide a detailed README.md file with instructions to run and test the application. It would also be recommended to provide a deployed demo (website, apk, etc).
3. The sample data/document/pictures will be provided for a few sections. Scrape data from TSG, KGP and related websites wherever required and not provided.
4. Paid Services can be opted for wherever required, but its demo version should be used for the competition.
5. Participants should keep in mind the access table while developing their application.
6. The application should be accessible to anyone and everyone, but the restricted sections should be accessible only after the login.

Rules and regulations:-

1. Deliverables:-

   Preferable Code Uploading Platform: GitHub
   Hosting Platform: Flexible
   Technologies: Flexible
   Suggestions:-
   For Frontend : React/Flutter
   For Backend : NodeJS/Python

2. The deliverables of the participants must be shared under open source license(s).
3. Teams can use libraries, frameworks, or open-source code in their projects, but acknowledging the source is a must.

4. Use of unethical sources is not allowed and may lead to disqualification.
5. Team strength of 2-12 members is allowed.
6. The hackathon will be conducted in 3 rounds:
   a. In round 1, the teams are expected to submit the architecture of the website.
   b. In round 2, the progress of the teams shall be considered
   c. In round 3, the teams are required to submit their final submissions of the application.
7. The number of points for each feature is fixed. Only the top three teams are awarded the points for each feature. Points division for First, Second, and Third will be done in the ratio of 5:3:2 of the maximum points.

   E.g., Let's say a feature has 50 points, then the points are awarded as 50,30,20 for first, second, and third positions, respectively.
8. In the case of a tie, an average of the points of tied positions will be given to each of the tied teams. E.g., Let's say two teams got second in a particular feature of 50 points, then each of them will be given 25 pts (30+20/2), and there will be no third position.
9. Each feature has the prize money fixed based on the number of points. The top team for a particular feature will be awarded the prize money. Each team can work on as many features as they can.
10. In every feature, a member of the top team may become a part of the Developer's society, IIT Kharagpur.
11. The teams with the highest number of points (points of the individual features added up) are awarded the overall first, second and third positions accordingly.
12. After the architecture round as well as the progress round, judges will provide feedback to each team.
13. Submission and feedback details shall be conveyed through the mail.

Round Details:-

First Round (Architecture Round)-

The teams have to submit a report in this round containing the architecture of their application. Some of the suggested components are:-

1. Technologies used for the frontend and backend.
2. Details of the features the team is planning to work on and presenting a tentative timeline for the same.
3. Front-end Prototype (may take the help of applications like Figma).
4. Specify Modals/Schemas for the backend.
5. Architecture explaining details from storage of data to delivering on the frontend.

Report Format:

- ○ Page Size: A4
- ○ Font Family: Times New Roman
- ○ Font Size: 12 (minimum)
- ○ Line Spacing: 1.15 cm (minimum)
- ○ Margin: 1 inch on all sides (minimum)

Second Round (Progress Round)-

In this round, the progress of the teams will be judged based on the tentative timeline they presented. Further instructions related to it will be conveyed through the mail.

Third Round (Final Submission Round)-

In this round, the teams will have to submit their final application files/codes.

Evaluation Parameters:-

The submission should follow the below criteria:-

- Scalability, practical use potential and viability in the given context.
- Relevance of pain points that the candidate's idea is trying to address.
- Ease of development (documentation, popularity, etc.) in the technology that is used.

- There should be no hardcoding of values, and data should be easy to update and serve dynamically from the backend.
- Separation of responsibility and modularisation of the feature(s) and different components.

Feature-wise marks

Feature	**Marks**
Home Page	120
Student Profile	100
Events (TSG)	100
Society Point	100
Students’ Point	100
News Bulletin	60
Quick Info	120
Archive	60
Notifications	120
Innovative Feature	120