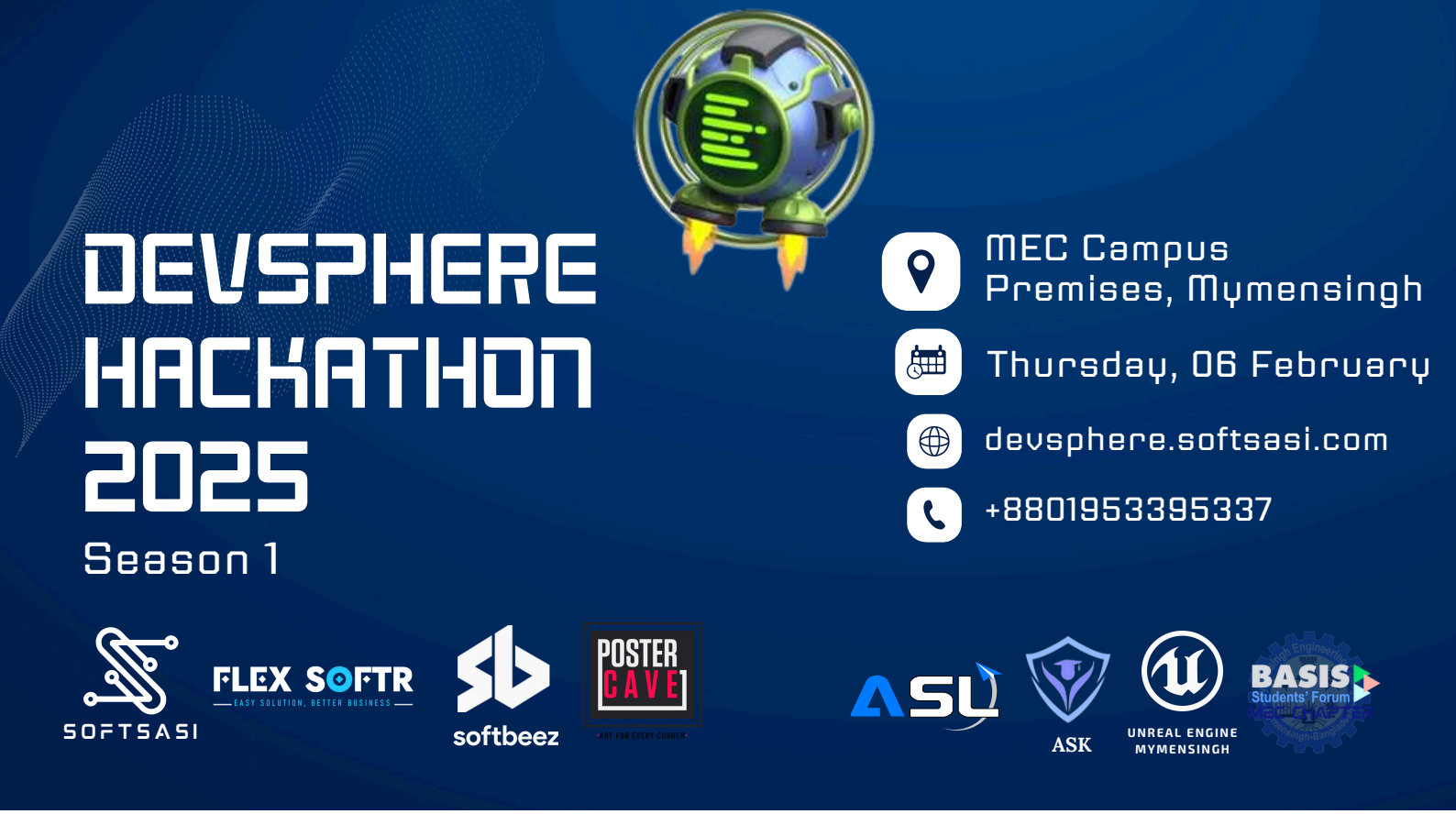

Smart Class Scheduler

Devsphere hackathon 2025
Inter-University Hackathon
Problem Description

Problem Statement

Alpha Science Lab faces challenges in scheduling knowledge-sharing sessions due to conflicts between mentor availability and member commitments to academic and club activities. The lack of a centralized scheduling system leads to ineffective class planning and low attendance.

Your challenge is to develop a **Smart Class Scheduling System** that allows members to mark their availability and helps mentors schedule sessions at optimal times to maximize participation.

Core Features

User Roles

1. **Members**
    - Have individual accounts.
    - Mark their daily availability on a calendar.
    - Enroll in different disciplines (courses).
    - Can look at their enrolled courses and upcoming sessions.
2. **Mentors**
    - Have individual accounts.
    - View availability insights of enrolled students.
    - Schedule sessions at optimal times.
    - Manage and update class schedules.
3. **Course Planner (Club President)**
    - Design, create, and delete courses.
    - Assign mentors to disciplines.
    - Manage overall scheduling and coordination.
    - Have an exclusive dashboard to oversee the system.

System Requirements

1. Separate Frontend & Backend:

- Build the front end (user interface) and back end (server logic) **independently**.
- You can Use **RESTful APIs** (standard web APIs) for communication between the two.

2. Backend Flexibility:

- Use **any backend language/framework** you're comfortable with (.NET, Java, Node.js, etc.).
- Work with **any database** (SQL: MySQL, PostgreSQL *or* NoSQL: MongoDB, Firebase).
- Add **basic user authentication** (e.g., login/logout) and **role-based access** (member/mentor/admin).

3. Frontend Options:

- Build a **web app** (React, Angular, Vue.js) **or mobile app** (Flutter, React Native, etc.).
- Create a **simple dashboard** where:
  - Members can mark their availability.
  - Mentors can view availability and schedule sessions.
  - Admins can manage courses and assignments.

4. Focus on Core Features First:

- Start with basic scheduling and availability features.
- Use mock data if needed (no need for complex databases initially).

Evaluation Criteria

- **Creativity (20%)**: Innovative implementation of scheduling algorithms, unique approaches to availability tracking, and novel solutions for session management.
- **Design (20%)**: Intuitive and easy-to-use interface.
- **Technical Implementation (30%)**: Proper backend-frontend separation, API design, and database efficiency.
- **Scalability & Security (10%)**: Handling of authentication, role-based access control, and data consistency.
- **Learning (20%): Demonstrated** growth in technical skills, effective use of new tools/frameworks, and documentation of challenges overcome during development.

Expected Deliverables

- **Fully functional backend and frontend** (connected via APIs).
- **Live demo** of the working system.
- **Source code repository** (GitHub or similar).

Wishing you all the best. 🚀✨