GLOBAL
ROBOTICS
CHALLENGE

RULE BOOK 2026

GLOBAL ROBOTICS CHALLENGE

MOBILE APP CHALLENGE

Smart Programmers
CHALLENGE

For More Information:

Scan QR Code

1. Technical Introduction :

- **The Mobile App Challenge aims to hone students' skills in software innovation by designing and developing practical applications using the MIT App Inventor platform, a simplified visual programming environment based on blocks programming instead of complex text code.**
- **In this competition, participants learn how to turn their creative ideas into real applications capable of meeting the needs of users or solving simple life problems, combining the technical and creative aspects.**
- **The competition also gives students the opportunity to express their ideas through educational, entertainment, health, or community applications, while enhancing logical thinking, programming, and teamwork skills within small teams.**
- **This competition is a first step towards entering the world of programming and application development, as it provides a hands-on and fun learning environment that shows students how simple ideas can be transformed into innovative digital solutions that serve society and facilitate people's lives.**

2. Team Composition :

- **Number of team members: 2 to 4 Students, guided by a Coach.**
- **Age Group:**

  Junior Category :

  - **Ages: 8 to 12 years.**

3. Areas and Challenges of the Competition:

- **Developing educational applications.**
- **Simple entertainment apps or games.**
- **Health and sports applications.**
- **Apps that serve the community or contribute to solving a real-world problem.**
- **Any app that aims to make a user's life easier or improve a daily experience.**

4. General Rules:

- **Each participant must bring their own laptop.**
- **The code/task must be delivered before the specified time (countdown).**
- **It is strictly forbidden to use the internet during the time of the competition.**
- **After each round, the team presents their work to the judging panel.**
- **The judging panel reviews the submitted code manually and evaluates it according to specific criteria.**
- **Before the start of each round, 15 minutes will be set aside to explain the challenge and answer the teams' questions.**
- **Any contact with people outside the team during the competition time is strictly prohibited.**
- **Any intervention or assistance from coaches/supervisors during the time of the rounds will result in a first warning, and its repetition may result in the team being disqualified from the competition.**

5. Rules for Scratch Programmers :

- **Age group allowed to participate: 8 to 12 years old.**
- **MIT App Inventor Allowed Only.**
- **The competition consists of 3 rounds, and each round is a different challenge explained by the moderator during the 15 minutes allotted before the start.**
- **Duration of each round: one hour only.**
- **After the time expires, what has been done is delivered without any further adjustment.**
- **After each assignment, the supervisor can ask questions to make sure students understand their code.**

6. Technical Requirements:

- **Junior Level:**

  Participants must have a basic knowledge of the following concepts in the MIT App Inventor:

  ❖ **UI Components:**

  These are the tools and items that appear on the application screen such as:

  - **Buttons: To execute commands when pressed.**
  - **Text Boxes: To enter data from the user, such as typing a name or number.**
  - **Images: To add graphic elements or logos to the app.**
  - **These components are the foundation for building the look of the app and the way the user interacts with it.**

❖ Event-Driven Programming:

It means that the app interacts with the user when a certain thing happens such as:

- Press a button.
- Enter text in a text box.
- Go to a new screen.
- That is, the app does not work on its own, but waits for an event from the user to execute the required codes.

❖ Variables & Logic Blocks:

- Variables: Used to store temporary data that the app can handle (e.g., storing a username or the result of a calculation).
- If/Else blocks: To check certain situations, such as: "If the number is greater than 10: Display a message."
- Loops: To repeat certain commands multiple times, such as displaying a message more than once or performing a repetitive operation.

❖ Navigation:

Most apps aren't limited to a single screen, so the participant needs to know how to:

- Create more than one screen within the app.
- Move from one screen to another (such as going from the sign-in screen to the home screen).

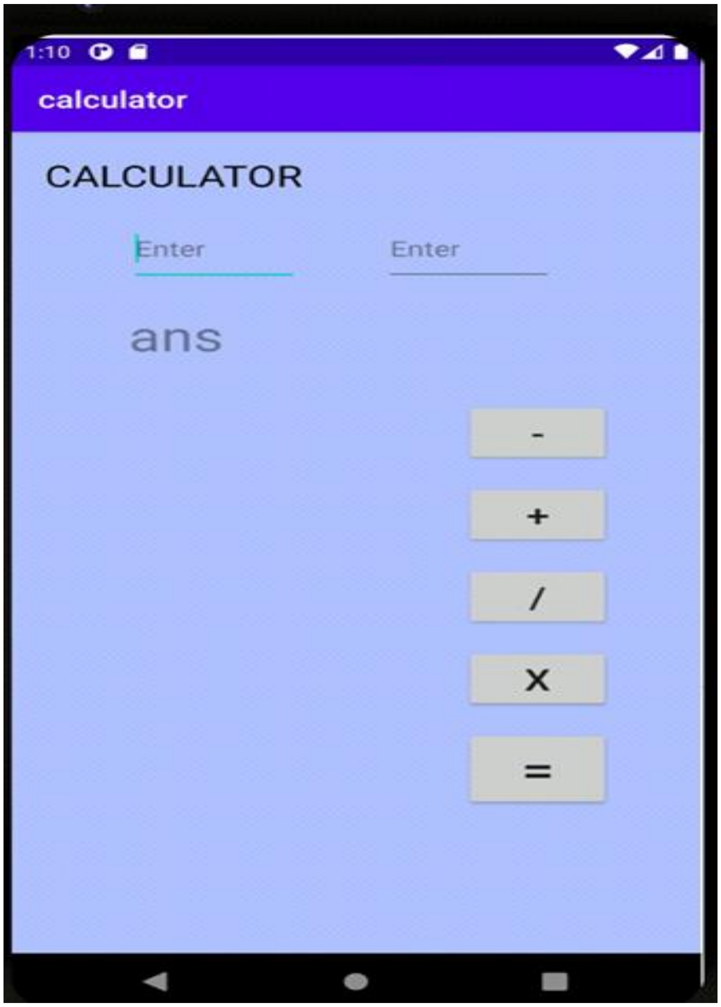
7. Guiding Examples :

- **Junior Level:**

Simple Calculator Application:

Function: Perform basic calculations (addition, subtraction, multiplication, division).

Mechanism: Enter two digits , choose the proc , and then display the result on the screen.



Good Luck.