



GLOBAL
ROBOTICS
CHALLENGE



RULE BOOK 2025

GLOBAL ROBOTICS CHALLENGE

WEB DEVELOPMENT



Smart Programmers
CHALLENGE



For More Information:

Scan QR Code

© COPYRIGHT
All Rights Reserved

1. Technical Introduction :

The website development competition aims to enhance students' skills in designing and developing web pages in a practical and competitive manner, enabling them to turn ideas into interactive websites with a well-designed and outstanding user experience. The competition is based on the principle of grading challenges according to age and experience, where the participants' abilities in using HTML, CSS, and JavaScript techniques to build real pages and websites are tested, with a focus on:

- Organize the code based on descriptive HTML elements.
- Designing responsive interfaces that work efficiently on various devices.
- Creativity and innovation in presenting the idea.
- Accuracy and adherence to technical standards.

2. Team Composition :

- **Number of team members:** 2 to 4 Students, guided by a Coach.
- **Age Group:**
 1. Junior Category :
 - ❖ **Ages:** 10 to 12 years.
 - ❖ Only HTML + CSS is allowed, with properties such as Flexbox and Grid applied.
 2. Senior Category:
 - ❖ **Ages:** 13 to 17 years old.
 - ❖ It allows the use of HTML + CSS with the application of properties such as Animations and Transitions. and media query
 3. Adult Category :
 - ❖ **Ages:** 18 years and above.
 - ❖ HTML + CSS + JavaScript is allowed without external libraries .

3. Areas and Challenges of the Competition:

- **Junior Category:** Designing a section **from a website** or **drawing**.
- **Senior Category:** Design a section **of a website** or **graphic** with the application of advanced CSS properties .
- **Adult Category:** Develop a small web application that combines HTML + CSS + JavaScript (e.g., calculator, To-Do List).

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 Website Developers:

- **Age group allowed to participate:** 10 to 18+ years old.
- Any code editor (IDE) such as: VS Code, Sublime Text is allowed
- 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.
- The challenges are broken down by age group:
 - ❖ **Junior Category:** Create a section **that displays products** or **graphics using HTML and CSS**.
 - ❖ **Senior Category:** Create a section **that displays products** or implement a drawing with the addition **of Animations** .
 - ❖ **Adult category:** Create a small web app (**such as a calculator**).
 - ❖ The use of the internet during the tour **is not allowed** to prevent cheating.

6. Technical Requirements:

- **Junior Level:**

- ❖ Master basic tags in HTML (headings, paragraphs, images, links, lists).
- ❖ Understand basic CSS properties (colors, backgrounds, borders, text alignment).
- ❖ Understand the Box Model (Margin, Padding, Border).
- ❖ Understand the transform property and pseudo elements.
- ❖ Semantic HTML <header>, <section>, <footer>.
- ❖ Knowledge of the principles of positioning (relative, absolute).
- ❖ Ability to use Flexbox to distribute items.
- ❖ Create simple shapes and graphics (circle, square, triangle) using CSS.

- **Senior Level:**

- ❖ All of the above are junior levels .
- ❖ Fundamentals of Responsive Design Media Queries.
- ❖ Positioning (relative, absolute, fixed, sticky).
- ❖ Use Flexbox and Grid to create organized layouts.
- ❖ Added Animations & Transitions to improve the user experience.
- ❖ Adherence to good design standards in terms of distribution and alignment.
- ❖ Dealing with Forms.

- **Adult Level:**

- ❖ All of the above junior and senior levels.
- ❖ Take advantage of CSS Variables.
- ❖ Create composite graphics using HTML + CSS.
- ❖ Add interactions using JavaScript:
 - Control of Page Elements (DOM).
 - Handling Events.
 - Form Validation.
 - Create a Responsive Navbar.
 - Use Local Storage to store data.

7. Guiding Examples :

- **Junior Level: Snowman Drawing**

- **Objective:**

Implement a snowman drawing using HTML and CSS only, while adhering to the selected colors.

- **Requirements:**

- ❖ Use HTML and CSS without images or SVG.

- ❖ Head and body design using white circles.

- ❖ Add eyes, nose, buttons, hands (twigs), hat, and scarf.

- ❖ Relying on characteristics such as:

- border-radius to create circles.

- Position and transform to arrange the items.

- before and after to add additional details such as the hem of the hat or the tail of the scarf.

- ❖ Write structured and clear code.

- ❖ **Required Colors:**

- ❖ Page Background:
#9ed2ec

- ❖ Face & Body: #ffffff

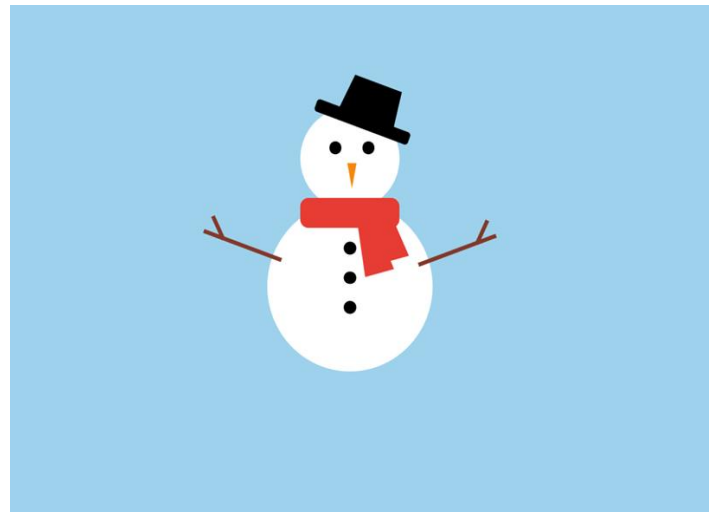
- ❖ Eyes: #000000

- ❖ Nose: #f7880f

- ❖ Hands: #84382a

- ❖ Hat: #000000

- ❖ Scarf: Red #e53935



❖ **Senior Level:** Login Form

❖ **Description:**

The participant is required to design a simple login form using only HTML and CSS. The form contains:

- Input fields (Email + Password).
- Login button.
- Help links such as "Forget Password" and "Register".

❖ **Design Preparation:**

1. **Background:**

- ❖ Gradient background of #4e8dd1 → #487fd2.

2. **Container:**

- ❖ A box in the middle of the screen.
- ❖ Semi-transparent background color is blue.
- ❖ Rounded edges. (Border Radius)
- ❖ Light Shadow (Box Shadow).
- ❖ Organize internally with Flexbox vertically.

3. **Texts:**

- ❖ Color: White.
- ❖ Title: (Login) in bold font.
- ❖ Secondary texts in a smaller font.

4. **Input Fields:**

- ❖ Two fields (Email + Password) with an underline instead of the frame.
- ❖ Small icons within fields (✉ for mail, 🔒 for password).
- ❖ Placeholder is light gray.

5. **Buttons:**

- ❖ Home button (login) in white and text in black.
- ❖ Full round edges. (50px)
- ❖ Hover effect.

6. **Links:**

- ❖ "Remember me" checkbox on the left.
- ❖ "Forget password?" On the right.
- ❖ Below the form: "Don't have an account? Register"

Login

Email



Password



☐ Remember me

[Forget password?](#)

Login

[Don't have an account? Register](#)

❖ **Adult Level: Calculator**

❖ **Description:**

Build a simple calculator using HTML+CSS+JavaScript.

❖ **Design Rules:**

1. **Background:**

- ❖ Background Color: #4e8dd1.
- ❖ Positioning the calculator using Flexbox .

2. **Container:** (Container)

- ❖ Color: White.
- ❖ Display: 350px .
- ❖ Rounded edges (10px).
- ❖ Light shade.

3. **Display :** (Display)

- ❖ Light grey background.
- ❖ Text in black.
- ❖ 24px font size.
- ❖ Align the text to the right.
- ❖ Height : 60px .

4. **Buttons:** (Buttons)

- ❖ Distribute using Grid (4 columns × 5 rows).
- ❖ Types of Buttons:
 - Numbers (0–9) with a light gray background.
 - Operations (+ - × ÷) in blue #487fd2.
 - Button (=) in #2ecc71 green color
 - The (C) button is #e74c3c red.
- ❖ Impact when passing and pressing.

❖ **JavaScript Functions:**

- Enter numbers and display them on the screen.
- Perform calculations (+ - × ÷).
- button (=) to show the result.
- (C) button to erase all data.

❖ **Expected Result:**

- In-text calculator, in white with a shadow and rounded edges.
- Light grey display.
- Numbered and organized buttons, their colors are fixed by type (numbers/operations/scan/equal).
- The calculator is working:
(enter numbers – run operations display result – erase).

Good Luck.