INNOVERSE

INNOVERSE
AI & Web Challenges
Guide

Https://innoverse.world
info@innoverse.world
challenges@innoverse.world

INNOVERSE

Overview

This guide provides detailed instructions for creating a programming challenge section for the Innoverse Invention & Innovation Expo. This section will include a downloadable PDF guide that outlines the challenges, instructions for participants, and guidelines for submitting their solutions via GitHub

Introduction

Welcome to the Innoverse Invention & Innovation Expo Challenges! This event aims to develop creativity, problem-solving, and technical skills among participants. In this guide, you will find everything you need to participate in our programming challenges, which are designed to be both fun and intellectually stimulating

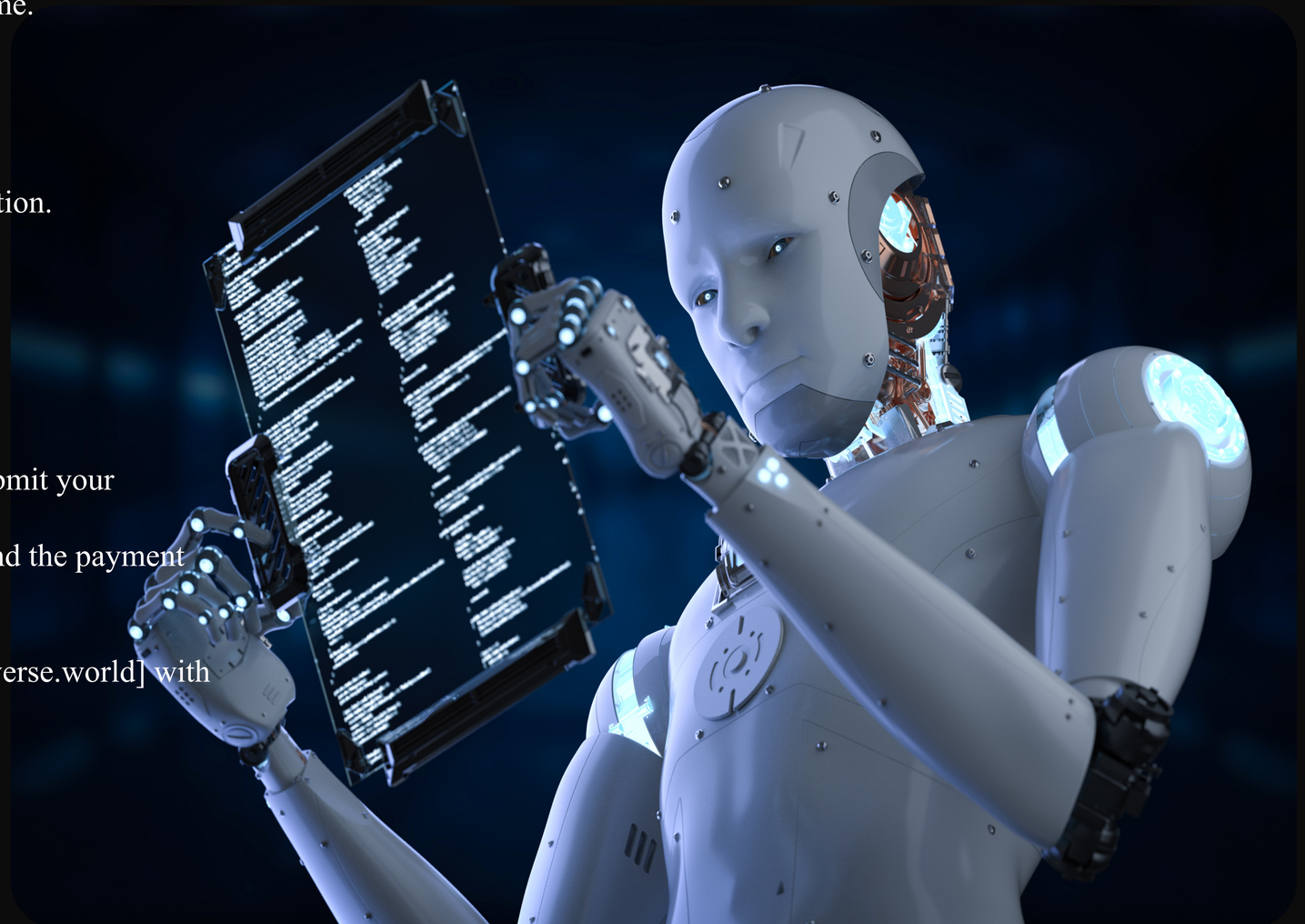
How to Participate

1. Register for the Innoverse Invention & Innovation Expo.
2. Download this PDF guide and read through the challenges.
3. Solve the challenge by writing code.
4. Upload your solution to GitHub.
5. Email us the link to your GitHub repository at [challenge@innoverse.world].

INNOVERSE

Submission Guidelines

1. **Create a GitHub Repository**
   - Create a new GitHub repository for your solution.
   - Name your repository in the format: ExpoChallenge_YourName.
2. **Upload Your Code**
   - Upload your code and any related files to the repository.
   - Ensure your code is well-structured and organized.
3. **Add a README File**
   - Include a README.md file in your repository for documentation.
   - The README should contain:
   - A brief description of your solution.
   - Instructions on how to run your code.
   - Any dependencies or libraries required.
   - Your contact information.
4. **Include a Payment Receipt**
   - Attach the receipt for your payment in the email when you submit your repository link.
   - Ensure the receipt is clearly legible and includes your name and the payment amount.
5. **Submit Your Solution**
   - Email the link to your GitHub repository to [challenge@innoverse.world] with the subject line "Programming Challenge Submission."
   - In the email body, include:
   - Your full name.
   - A brief summary of your approach to solving the challenge.
   - Attach the payment receipt.



INNOVERSE

Judging Criteria

Innovation & Creativity 30%

Technical Execution 30%

Impact & Usefulness 20%

Presentation & Clarity 20%

INNOVERSE

1. Cyber Warehouse of Defective Robots (Web)

In the year 2125, Earth is run by the robot manufacturing company CyberCore. Thousands of defective robots are kept in huge warehouses. Your task is to build a management system to determine the storage location of each robot in the warehouse in such a way that the warehouse space is used optimally.
Challenge goal:
Using JavaScript and HTML/CSS, design a visual system that places the robots as rectangles with different sizes in a two-dimensional grid, in such a way that:
• No two robots collide with each other
• The empty space between them is minimized
• The robot data is received via a JSON input
Sample input (JSON):

```
[
{ "id": "robot_01", "width": 2, "height": 1 },
{ "id": "robot_01", "width": 2, "height": 1 }
]
```

Rules and output:
• The output must be a web page that displays the location of each robot in the grid.
• The x and y position of each robot in the grid must be displayed.
• The arrangement must fit in a 10x10 space (or larger depending on the number of robots).

iNNOVERSE

2. Smart City and Traffic Light Management with Queue Algorithm(Web)

Design a smart traffic light system at a busy intersection using queue, prioritization, and automatic control in the browser.
Challenge goal:
Simulate a 4-way intersection in a smart city, where you must manage the traffic lights using JavaScript and queue algorithms and dynamic decision-making, so that:
• Traffic is optimized
• Cars move based on a specific priority
• The system automatically schedules the lights and decides which path should open.
Story scenario:
You are responsible for designing the digital brain of an intersection in the smart city "Innoverse." In this city, cars enter the intersection every few seconds and if the signaling system is not proper, traffic gets locked.
In this challenge you must design a system that can:
• Detect in real time which path has more traffic.
• Decide which path should be open and the others should stay red.
• Let the cars pass in order and based on priority.
[h , h , h , 0 , h , h , h , h]
[h , h , h , 1 , h , h , h , h]
[0 , 0 , 1 , 4 , 1 , 1 , 0 , 0]
[h , h , h , 1 , h , h , h , h]
[h , h , h , 1 , h , h , h , h]

3. Hidden Path Decoder (Web)

Build an interactive game based on discovering a path in a graph grid with specific rules.
We have a square grid (like a 10×10 chessboard), where each cell can be one of the following:

- Free path (with white color)
- Obstacle (with black color)
- Start point (green)
- End point (red)

The player can only try to find a correct path from the green cell to the red cell by clicking on cells, without clicking on obstacles. But that's not all:
Main rule:
The participant must design a system in the browser that:

1. Builds the map from a predefined two-dimensional array (for example, a JavaScript matrix where value 0 means path, 1 means obstacle, 2 means start, 3 means end).
2. Allows the user to click on the cells and build his/her path step by step.
3. After each click, the selected path is shown in a special color (for example, blue).
4. Finally, if the selected path:
   - Has only passed through valid cells
   - Has reached from the start point to the end point
   - Has not been repetitive
   - And has not hit any obstacles

   Otherwise, an error or rejection message is displayed.

INNOVERSE

4. Encrypting Messages as Visual Codes in the Browser (Web)

In this challenge, you must design a system that can convert a simple text message (for example, a short sentence or a phrase) into an encrypted image. This image can only be decoded with specific rules.
All encryption and decryption steps must be done in the browser, and there is no need for backend or external files.
Main goal:
Build a tool in the browser that has two main functions:

1. Encrypt text to image
The user enters text (for example: HELLO). The system, using a defined algorithm, converts it to an image consisting of blocks (for example, squares with specific colors or special positions).
Each character must be converted into a unique graphical pattern. For example: H = red square at the top left, E = blue circle in the center, and so on.

2. Decrypt image to text
The user can give the image generated by the system back to the system (or even "read" it by clicking on a graphic map), and the system must reconstruct the original text.

Rules and limitations:

- Each character from A-Z must have a specific "graphical pattern" (for example, a combination of color + position or a special shape).
- These patterns must be dynamically generated in the code, not pre-designed images.
- Only div, span, and CSS can be used.
- The decoding system must "read" the image exactly based on these rules.

1. Saving the Ecosystem with AI (AI)

In the near future, global agricultural systems operate based on weather, soil, and water resource data. Governments have asked you to build a model that predicts which regions are at high risk of severe drought.

Data:

- A set of climate data (temperature, precipitation, humidity)
- Water resource data and the cultivated area of each region

Task:

- Design a model in Python (using scikit-learn or TensorFlow) to predict drought
- Build a simple dashboard using Streamlit or Gradio to display regions at risk

INNOVERSE

2. Virtual Avatar with Your Personality (AI)

In the year 2035, digital interactions have reached a level where people can create a virtual version of themselves. These avatars not only simulate your appearance and voice but can also think and react on your behalf. In the future world, having a personal avatar for business meetings, customer support, or even as a virtual partner is considered a necessity.

Challenge goal:
Design an AI system that can:
- Build a digital avatar that simulates your appearance and voice
- Naturally respond to questions and sentences as if you were conversing yourself
- Be able to connect to websites or social media platforms to act as your digital representative

Suggested technologies:
- Voice cloning: to create a user-like voice (e.g., using models like VITS or ElevenLabs)
- Character LLM fine-tuning: to mimic the user's thinking style and speaking manner (e.g., fine-tune a large language model like LLaMA or GPT based on personal conversation data)
- Avatar animation: to create animated avatars (e.g., using Unreal Engine or Avatar SDK)

Rules and output:
- The output must be a visible and audible virtual avatar (video or interactive environment)
- The avatar must respond to at least 10 simple questions, and the answers must match the user’s style
- The avatar’s appearance and voice should be as realistic and simulated as possible

3. Emotion Painter (AI)

In the future world, humans use art as a primary tool to express their inner emotions. An art startup has decided to create a system where each person can create a unique artwork simply by saying or writing a sentence that expresses their inner feeling.

Challenge goal:
Build an AI system that:
- Receives user input (text sentence or audio file) and analyzes the emotions present in it
- Based on the identified emotions, creates a digital image representing those feelings

Suggested technologies:
- Emotion classification: analyze emotions from text or audio (e.g., using BERT for emotion detection or audio models like Whisper or Wav2Vec)
- Generative models: create images with DALL·E or Stable Diffusion based on final text or emotion tags
- Frontend interactive: build a simple app or dashboard (Streamlit or Gradio) for testing and displaying

Rules and output:
- The output must be a digital artwork that is downloadable or shareable
- The system must be able to detect at least 5 different emotions (e.g., joy, sadness, anger, fear, calmness) and generate an image accordingly
- Input should be accepted from both text and audio

INNOVERSE

4. Smart Productivity Camera (AI)

In the future, large companies need to precisely monitor their employees' productivity. A security company wants to build a system that can determine whether a person is working effectively or not, solely by analyzing images and video.

Challenge goal:
Design an AI system that can:
- Analyze employee behavior in the workplace through live video or images
- Detect different states (productive work, leaving the workstation, sitting idle, moving between desks, etc.)
- Display analytical data in a dashboard

Suggested technologies:
- Computer Vision: to analyze video and frames
- Pose Estimation: to detect body posture (e.g., MediaPipe or OpenPose)
- Action Recognition: to detect activity type from video (e.g., I3D or SlowFast)
- Dashboard frontend: to graphically display results (e.g., Streamlit or Plotly Dash)

Rules and output:
- The output must classify each individual in at least 3 different states
- The system should be able to process a short video (30 seconds to 1 minute) and generate an analytical report at the end
- The dashboard must show the percentage of productive versus unproductive time

INNOVERSE

Need Help!?

We are here to support you!

info@innoverse.world

www.innoverse.world