Google Developer Student Club

Inductions 2025

Deadline: 16th February 2025 (Sunday) 11:59 PM IST

Instructions:

- ***You must choose only a single task from the options provided.***
- ***This is an individual task; collaboration leads to immediate disqualification.***
- ***Create a GitHub account here.***
- ***Good Coding Practices: Naming Conventions, Meaningful Variable and File Names, Appropriate Folder Structure, Code Readability, and Comments are encouraged.***
- ***Each task is self-contained; interpretations may vary, and there is no one "answer" to a single problem.***
- ***Make sure to add a README file in your GitHub repository consisting of:***
  - ***Brief task description***
  - ***Related screenshots***
  - ***Instructions to run the project***
  - ***If hosted, its URL***
- ***Submit the code even if the task is incomplete.***
- ***Create a public repository for your task and submit the link in the form.***
- ***Task prioritization underscores the emphasis on challenging assignments, with evaluation leaning towards recognizing and rewarding diligent efforts in handling more intricate tasks over simpler ones.***

For any queries, contact:

Yash (+91 93731 82023)
App Development

Yash Singh (+91 95609 67016)
Blockchain

Divyam (+91 99099 77177)
Game Development

App Development

Task-1: E-COMMERCE APPLICATION

Problem Statement:

Develop an e-commerce application that evolves through three progressively advanced levels. The evaluation criteria will be based on UI design, code quality, and functionality.

LEVEL 1- UI-Heavy Implementation

- Focus primarily on an intuitive and visually appealing user interface.
- Clean, structured, and maintainable code is essential.
- Judging Criteria: UI aesthetics and code quality.

LEVEL 2- Functional App with Local Backend

- Implement a fully working application with a dummy backend or local database.
- Ensure seamless navigation and basic e-commerce functionalities.
- Judging Criteria: Application functionality and overall code quality.

LEVEL 3- Full-Scale E-Commerce Solution

- Integrate a fully operational backend with a remote database.
- Implement features such as authentication, payment processing, and order tracking.
- Judging Criteria: Comprehensive functionality, backend integration, and clean code.

Task-2: CALCULATOR APPLICATION

Problem Statement:

Develop a calculator application, progressively enhancing its capabilities through different levels.

LEVEL 1- Basic Calculator

- Implement fundamental arithmetic operations.
- Optionally, allow the app to remember previous values.
- Judging Criteria: UI quality and clean code.

LEVEL 2- Scientific Calculator

- Implement a fully working application with a dummy backend or local database.
- Ensure seamless navigation and basic e-commerce functionalities.
- Judging Criteria: Application functionality and overall code quality.

LEVEL 3- Calculus Calculator with API Integration

- Implement an advanced calculator capable of handling calculus functions.
- Utilize API calls for complex mathematical computations.
- Maintain history of previous calculations.
- Judging Criteria: Functionality, performance, and clean code.

Task-3: AI THEMED APPLICATION

Problem Statement:

Develop an AI-powered application such as a chatbot or an app inspired by Google Lens.

Requirements:

- The AI component should be central to the application's functionality.
- Demonstrate innovative use of AI/ML capabilities.
- Judging Criteria: AI integration, practical usability, and clean implementation.

Useful Resources:

- Flutter tutorial, Flutter documentation
- Kotlin
- JetPackCompose documentation
- Adding client dependencies
- Ktor-Client

Submission Guidelines:

- A one-page report detailing the app's development process, key features, and technical stack must be submitted alongside the application.
- Ensure clean and well-documented code.
- Submissions will be evaluated based on adherence to the defined levels, overall design, and implementation quality.


Blockchain

Task-1: ERC20 TOKEN DEVELOPMENT & DEPLOYMENT (Beginner Level)

Problem Statement:

Digital ecosystems often require a seamless way to reward users, facilitate transactions, or enable governance. Many Web3 applications use their own ERC20 tokens to address such needs.

Objective:

- Develop an ERC20 token with basic functionalities like transfer(), approve(), and transferFrom().
- Deploy the contract on a local Ethereum fork (using Hardhat, Foundry, or Ganache).
- Deploy it on a testnet (Sepolia, Goerli, etc.).

Resources:

OpenZeppelin ERC20 Documentation

Task-2: ERC721 NFT DEVELOPMENT & DEPLOYMENT (Intermediate Level)

Problem Statement:

Ownership verification and digital uniqueness are critical challenges in various industries. Blockchain provides a solution through tokenized assets, enabling better traceability, proof of ownership, and authentication.

Objective:

- Develop an ERC721 token with metadata, ownership, and minting functionalities.
- Upload metadata and images to IPFS.
- Deploy the contract on a testnet (Sepolia, Goerli, Mumbai, etc.).

Resources:

OpenZeppelin ERC721 Documentation

Task-3: SMART CONTRACT TESTING (Advanced Level)

Problem Statement:

Blockchain applications require high reliability and security. Proper testing ensures contract functionality, security, and efficiency.

Objective:

- Write comprehensive tests for ERC20 and ERC721 contracts.
- Test transfers, approvals, minting, ownership changes, and other key functions.
- Use a development environment of your choice:
    - **Hardhat** → JavaScript
    - **Foundry** → Solidity
    - **Brownie** → Python

Final Deliverables:

1. ERC20 and ERC721 smart contracts
2. Deployment on testnet and local fork
3. IPFS metadata for NFTs
4. Test cases and execution reports

Game Development

Task: GAME JAM IN ITCH.IO

Problem Statement:

Make a game that takes and implements 2 out of the following 6 themes:-

- I am Invincible
- Twists and Turns
- Leaks
- Every Life has a Price
- Flow
- Mind your Balance

NOTE: These are some open ended themes which can be incorporated in the games in a number of ways. For example, for any game with a theme "Interconnected networks", one can make a series of environments which interact with each other to influence changes. Use your own creativity to induce the features in such a way that the target number of features are met.

Resources:

1. Unity tutorial
2. Unreal Tutorial

Google Developer Student Club, IIT Indore