

Please choose one 🌟 INSTRUCTIONS FOR ALL TRACKS:

- * Submit your completed project as a *GitHub repo* or *Google Drive link*.
- * Also submit a *2–5 minute video demo*:
 - * Show the *working frontend*.
 - * Explain your *code structure and choices*.
 - * Mention anything you would improve with more time.
- * You have *24 hours* from receiving the task.

🖥️ TRACK 1: PHP DEVELOPER (Frontend + Backend)

💡 Task:

Build a *Task Manager Web App* with a *PHP backend* and *HTML/CSS/JavaScript frontend*.

Features:

- * Users can:
 - * Create, edit, and delete tasks.
 - * Mark tasks as “To Do,” “In Progress,” or “Done.”
 - * View tasks in a list with status color indicators.

Requirements:

- * Backend: *PHP 8+* (Laravel, Symfony, or plain PHP).
- * Frontend: HTML/CSS + Vanilla JS or basic framework (Bootstrap encouraged).
- * No login required, but store tasks in a file or database (SQLite/MySQL).

Bonus:

- * Add *drag-and-drop task sorting*.
- * Make it *mobile responsive*.

📱 TRACK 2: ANDROID DEVELOPER (Frontend Focused)

💡 Task:

Build a *News Reader App* with a polished user interface.

Features:

- * Fetch and display articles from a public API (e.g., NewsAPI.org or JSONPlaceholder).
- * Users can:
 - * Scroll a list of headlines with images.
 - * Tap a headline to see the full article.
 - * Favorite an article (store locally).

Requirements:

- * Built in *Kotlin*.
- * Use *Jetpack Compose* or XML UI.
- * Use *MVVM* or other structured pattern.

Bonus:

- * Support *offline caching*.
- * Add *animations* on navigation.

📱 TRACK 3: FLUTTER DEVELOPER (Frontend Showcase)

♦ Task:

Build a *Shopping Catalog App* using Flutter.

Features:

- * Fetch product list from <https://fakestoreapi.com>.
- * Display products in grid view.
- * Allow users to favorite items (persisted locally).

Requirements:

- * Use *Flutter 3.x*.
- * Implement a clean folder structure (models, screens, widgets).
- * Use http package for API call and provider or setState for state management.

Bonus:

- * Add *category filtering*.

* Include *animations* or custom widgets.

🐸 TRACK 4: PYTHON DEVELOPER (Django/Flask + HTML)

♦ Task:

Create a *Notes Web App* using a Python backend with a simple HTML/CSS/JS frontend.

Features:

* Users can:

- * Create, edit, and delete notes.
- * See note list with timestamps.
- * Search notes by keyword.

Requirements:

- * Backend: *Flask* or *Django*.
- * Frontend: Serve a basic web interface with forms and table views.
- * Use SQLite or in-memory database.

Bonus:

- * Add a *dark mode toggle*.
- * Add a *search filter* using JavaScript.

📺 VIDEO SUBMISSION GUIDELINES

Length: 2–5 minutes

Content to cover:

- * Show the working UI.
- * Walk through main features.
- * Explain the project folder structure.
- * Share anything you'd improve if given more time.