Frontend Development Assignment: Option 1

Develop a YouTube Aggregated Free Skill Course Detail Page

Objective:
Develop a **static webpage** that showcases a course detail page for a YouTube Aggregated Free Skill Course using **HTML, CSS, and JavaScript**. The goal is to create a functional mockup that mirrors the design and enhances user experience through interactive and aesthetic elements.

Use Case:

The platform curates YouTube videos into structured, skill-based courses. Users should be able to:

- View course details like title, level, duration, number of videos, and ratings.
- Navigate through a list of videos with details such as title, duration, views, and completion status.
- Interact with features like AI-generated summaries, custom notes, and progress tracking.

Core Requirements:

1. Course Overview Section

- A prominent **"Enroll" button** at the top of the page.
- Display course details such as:
    - Title of the course.
    - Skill level (e.g Beginner, Intermediate, Advanced).
    - Estimated duration.
    - Total number of videos.
    - Average user rating (e.g 4.5/5).

2. Video List Section

- Display a scrollable list of videos with the following details:
    - Title.
    - Duration.
    - Number of views.
    - A “Mark as Completed” checkbox.
- Automatically **tick-mark completed videos** and allow navigation to the next/previous video.

3. Video Player Section

- Embed a YouTube player for playing videos.
- Show captions/subtitles for the video.
- Display an **AI-generated summary** based on video captions.

4. Notes and Tools Section

- Allow users to **add and save custom notes** for each video.
- Display **AI-generated questions** to test learning from the video.

5. Additional Features:

- **Progress Tracker:** A visual progress bar to indicate course completion percentage.
- **Job-Seeker Tips:** Add a section for sample job descriptions or career advice related to the course topic.
- **Rate the Course:** Allow users to leave a rating for the course.

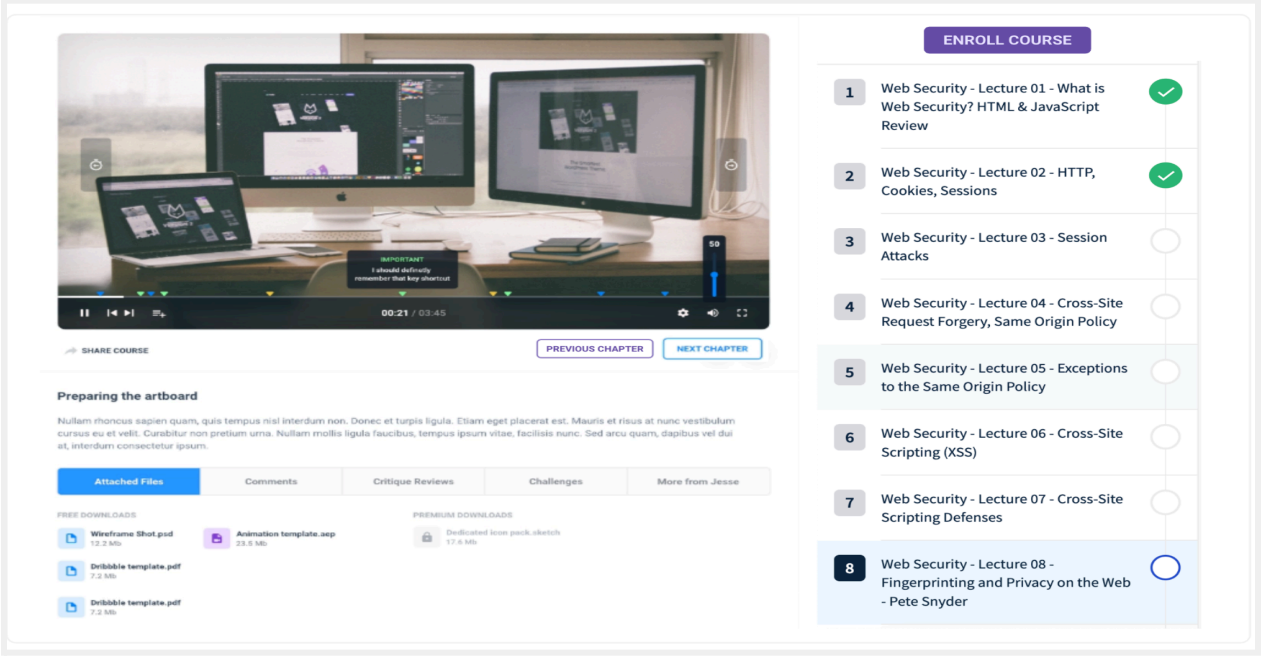
Bonus Points (Optional):

1. **Gamification:**Implement badges, points, or leaderboards for course engagement.
2. **Community Engagement:**Add a discussion forum or comment section for video-based discussions.
3. **Certificate of Completion:**Display a "Download Certificate" button for users who finish the course.

Deliverables:

1. **Submit the following files:**
    - **HTML file for the webpage structure.**
    - **CSS file for styling.**
    - **JavaScript file for interactivity.**
2. **(Optional) Include a readme.txt explaining your approach and any assumptions made.**
3. **Share a ZIP file of the project via email or a hosted live demo link.**

Note: The sample might have a few missing details. Use it as a reference, but do not limit your creativity or restrict your design ideas to what's shown in the sample. Feel free to enhance, improve, and include additional features or better design elements.



Frontend Development Assignment: Option 2

Mobility Aggregator Platform

Objective:
Develop a **static multi-page website** for a **Mobility Aggregator Platform** using **HTML, CSS, and JavaScript**. The goal is to create a visually appealing and functional static website that categorizes and displays popular mobility apps across top cities in India.

Use Case:

The platform should:

- Detect the user's current location (simulated in JavaScript for this task) and display relevant mobility apps.
- Allow users to select other cities via a dropdown and display apps for those locations.
- Show a placeholder message for cities without data: *"No data available for this city. We will be adding it soon!"*

Core Requirements:

1. Homepage (Mobility Aggregation Display)

- Group mobility apps under the following categories:
    - **Cabs:** Uber, Ola, Meru, etc.
    - **Bikes:** Rapido, Bounce, Vogo, etc.
    - **Autos:** Ola Auto, Jugnoo, etc.
    - **Porter Services:** Porter, Blowhorn, etc.
- Automatically display apps based on the simulated detected location (default to **Delhi** if no location is provided).
- Provide a **dropdown menu** to select other cities and view mobility apps.
- Display the placeholder message for cities with no data: *"No data available for this city. We will be adding it soon!"*

2. Engagement Features on Homepage

- **Antique Mobility Museum Section:**
    - Show trivia, images, or facts about antique cars, bikes, or mobility-related history for the selected city.
- **Popular App Icons:**
    - Display clickable logos/icons for popular apps (e.g., Uber, Ola) that redirect to external app links.
- **Call-to-Action Section (CTA):**
    - Add a **sign-up form** where users can subscribe to receive updates on upcoming mobility options.

3. Mobility App Detail Page

Create a detailed view page for each mobility app that includes:

- **Company Overview:** A short description of the company.
- **Download Links:** Provide Android and iOS app download buttons.
- **Booking Links:** Add a direct booking link (if available).
- **Ratings & Reviews:**
    - Display existing reviews (static data).
    - Allow users to submit feedback.

Bonus Points (Optional):

1. **Responsive Design:**
    - Ensure the website is fully responsive and adapts well to both desktop and mobile devices.
2. **Animations:**
    - Use CSS or JavaScript for hover effects, smooth scrolling, or transitions.
3. **Interactive Features:**
    - Simulate location detection using JavaScript and dynamically display city-specific data.
    - Add a visual rating system (stars or sliders) for user reviews.

Technical Expectations:

- Use plain **HTML, CSS, and JavaScript** (no libraries or frameworks).
- Focus on clean, structured, and well-documented code.
- Ensure proper styling with CSS for an aesthetically pleasing design.
- Simulate location detection and dynamic data updates with JavaScript.

Deliverables:

1. Submit the following files:
    - **Homepage:** Display categorized mobility apps and engagement features.
    - **App Detail Page:** Display detailed information for a single mobility app.
    - **CSS and JavaScript files:** For styling and interactivity.
2. (Optional) Include a **readme.txt** explaining your approach and any assumptions made.
3. Share a **ZIP file** of the project or a live demo link.

Frontend Development Assignment: Option 3

Redesign the LueinHire Candidate Page

Objective:
Create a static webpage of the **LueinHire Candidate Page** (https://lueinhire.com/candidate/) using HTML, CSS, and JavaScript. The page should be visually appealing, responsive, and functional, offering a

clean and engaging layout.

Assignment Scope:
You are required to implement a simple static webpage that includes the following key features:

1. **Core Platform Offerings:**
    - **Fresh Job Opportunities:** Showcase access to over 1M+ jobs for candidates.
    - **Practice Interviews:** Create an interactive space for interview preparation.
    - **Mock Live Interviews:** Simulate real-time interviews to build candidate confidence.
    - **Smart Resume Builder:** Incorporate a tool for building optimized resumes.
    - **Curated Online Courses:** Highlight access to 50,000+ free upskilling courses.
    - **Resume Review Service:** Allow candidates to submit resumes for professional review.
    - **Placement Training Program:** Present this comprehensive placement preparation service.
    - **Coding Challenges:** Include a section for solving tech coding problems.
    - **Quizzes:** Provide interactive skill-building quizzes to engage users.
    - **Job Recommendations:** Emphasize curated job recommendations tailored to the candidate's profile.
2. Feel free to incorporate **innovative ideas or features** to elevate the candidate experience further.

Design Guidelines:

- Prioritize a **clean, modern, and professional design** that aligns with the LueinHire brand.
- Ensure a **mobile-friendly, responsive layout** for seamless access across devices.
- Use intuitive navigation and a well-structured layout to enhance usability.
- Highlight calls-to-action (CTAs) prominently to encourage user engagement (e.g., "Start Mock Interview," "View Jobs," "Submit Resume for Review").

Deliverables:

1. Static webpage (HTML, CSS, JavaScript files).
2. Optional brief **readme** explaining your approach
3. Share a **ZIP file** of the project or a live demo link.