Design Considerations and Implementation Guidelines for PoD-Based Full Stack Development Projects - For ADM, DE-DX, ISG Tracks

1. PoD Organization

- **PoDs:** The cohort will be divided into **PoDs** (Pods of 5 GenCs), ensuring an effective team-based learning experience.
- **Project Allocation:** Each **PoD** will be assigned **one project**. The project should cover both **frontend** and **backend** development to ensure that all members of the PoD gain experience with both parts of the stack.
- **Module Ownership:** In each PoD, each member will be assigned a module within the project, with both **frontend** and **backend** responsibilities. This will ensure that each trainee is involved in full-stack development.
- **PoD Leader:** A **PoD Leader** will be selected from each PoD to:
  - Act as the point of contact for the **technical trainer**, **cohort mentor**, and **batch owner**.
  - Coordinate activities, facilitate communication, and address any concerns or blockers within the team.
  - Ensure that Agile practices are being followed effectively.
  - Take responsibility for the PoD's progress and ensure tasks are completed on time.

2. Implementation Methodology

- **Agile Methodology:** Each PoD will follow **Agile Scrum** methodology for development. The project will be broken into **Sprints**, typically lasting **2 weeks**.
  - **Sprint Planning:** At the beginning of each sprint, the PoD will define its sprint backlog, which will consist of tasks and features to be developed or completed during the sprint.
  - **Daily Standups:** The team will hold **daily standup meetings** to track progress, discuss challenges, and ensure smooth collaboration.
  - **Sprint Review:** At the end of each sprint, a **sprint review** will be held to demonstrate completed work and gather feedback from the team and mentors.
  - **Sprint Retrospective:** After each sprint, the team will conduct a **retrospective** meeting to reflect on what went well, what could be improved, and what actions need to be taken in the next sprint.
- **Backlog Management:**

- **Product Backlog:** The PoD Leader is responsible for maintaining the **Product Backlog**, which contains all the high-level requirements for the project.
- **Sprint Backlog:** The team will define their **Sprint Backlog** by selecting items from the product backlog that they aim to complete during the sprint.

3. Version Control and Collaboration

- **Private GitHub Repository:**
  Each PoD will maintain a **private GitHub repository** for collaborative development.
  - **Repository Setup:** The **PoD Leader** will initiate the GitHub repository and invite all team members to collaborate.
  - **Branching Strategy:** Use a **feature-branching** model, where each developer creates a new branch for the specific feature or task they are working on. Once completed, they will submit a **pull request** for review.
  - **Code Review:** All team members should review each other's code, ensuring quality and adherence to best practices.
  - **Merging Code:** Pull requests should only be merged into the main branch after thorough review and approval from at least one team member.
- **Security Policy Compliance:**
  - **No Organizational Data:** No organization-specific data or proprietary code should be uploaded to the repository, as this may violate company security policies. This includes not uploading sensitive information like access keys, passwords, or personal data.
  - **Access Management:** Ensure that only authorized team members are added to the repository. The PoD Leader should manage access.

4. Project Tracker and Development Progress

- **Development Tracker:**
  Coaches will maintain a **project tracker** to monitor progress. The tracker will include:
  - **Milestone Completion Dates**
  - **Current Sprint Tasks**
  - **Open Issues/Blockers**
  - **Code Review Status**
  - **Progress on Features**
- **Interim Evaluation (Individual Basis):**
  - An **interim evaluation** will be conducted halfway through the project. It will focus on evaluating the project's progress, including completed modules and features.

  - The evaluation will assess the backend functionality developed thus far, ensuring the project is on track and identifying any potential risks.
- **Final Evaluation (Individual Basis):**
  - The **final evaluation** will include a comprehensive review of the entire project. This will involve assessing both the **frontend** and **backend** modules for integration, functionality, UI/UX design, code quality, and Agile practices.
  - The project will be assessed based on its completion, usability, integration between different parts of the system, and the quality of the code.

5. Roles and Responsibilities

- **PoD Leader Responsibilities:**
  - Coordinate with team members to ensure tasks are completed.
  - Maintain the **Product Backlog** and define the **Sprint Backlog**.
  - Manage communication with the **technical trainer**, **cohort mentor**, and **batch owner** to resolve blockers.
  - Lead daily standups, sprint reviews, and retrospectives.
  - Ensure that team members follow Agile practices.
- **PoD Member Responsibilities:**
  - **Backend Development:** Implement APIs, business logic, and database management for the assigned module.
  - **Frontend Development:** Develop the user interface and integrate it with the backend.
  - **Testing:** Perform unit and integration testing to ensure the correctness of the implemented features.
  - **Collaboration:** Collaborate effectively with other team members, particularly when integrating backend and frontend components.
  - **Documentation:** Write clear and concise documentation for code, features, and API endpoints.
- **Trainer Responsibilities:**
  - Provide technical assistance during the sprint, helping the teams overcome blockers.
  - Conduct **code reviews** and offer feedback to improve code quality.
  - Monitor adherence to Agile principles and guide the PoDs through the implementation of Agile processes.
- **Cohort Mentor Responsibilities:**

    - Oversee the overall progress of the cohort and ensure that teams are working effectively.
    - Ensure proper application of Agile practices and address any issues regarding team collaboration.

- **Batch Owner Responsibilities:**
    - Ensure the successful organization and execution of the cohort's activities.
    - Oversee the PoD assignments and make sure that each team has the required resources.
    - Resolve any concerns raised by PoD Leaders and provide support to ensure smooth project execution.

6. Collaboration Tools

- **Communication Tools:**
  Use tools like **MS Teams**, or **Outlook** for team communication. Each PoD should create a dedicated channel/group to share updates, ask questions, and discuss blockers.
- **Project Management Tools:**
  Use **Excel** for managing sprint backlogs, task assignments, and milestones.

7. Coding Standards and Best Practices

- **Code Quality:**
  Follow best practices for writing clean, readable, and maintainable code.
    - Ensure the code adheres to **SOLID principles** for object-oriented design.
    - Follow **standard naming conventions** and provide meaningful comments for clarity.
- **Documentation (Optional):**
    - Write clear and concise documentation for each module, explaining the functionality and how to interact with the APIs.
    - Provide detailed instructions for setting up the development environment and running tests.
- **Testing:**
    - Implement **unit tests** for individual components of the backend and frontend.
    - Use **Postman** or **Swagger** to test API endpoints.
    - Implement **integration tests** to ensure the modules work together as expected.

8. Evaluation Criteria

- **Interim Evaluation:**

  - **Functionality:** Evaluate the completeness of the modules implemented so far.
  - **Code Quality:** Review the quality of the code, following standards and best practices.
  - **Collaboration:** Assess how well the PoD members are collaborating and following Agile practices.
- **Final Evaluation:**
  - **Complete Functionality:** The entire project should be fully functional, including all modules.
  - **Code Quality:** The code should be maintainable, well-documented, and follow coding standards.
  - **UI/UX:** The user interface should be responsive, intuitive, and meet design requirements.
  - **Testing:** All features should be well-tested, with both unit and integration tests.