Project Design Considerations and Implementation Guidelines

Design Considerations

1. **Team Composition:**
   - Teams (PoDs) consist of 5 members, each responsible for implementing a specific **module**.
   - Each member handles both **frontend** and **backend** development for their assigned module.
   - A **PoD Leader** is designated to manage communication and coordination with stakeholders.
2. **Agile Methodology:**
   - Follow **Agile development** to promote iterative progress and regular feedback.
   - Maintain a **Product Backlog** for overall requirements and a **Sprint Backlog** for sprint-specific tasks.
3. **Collaboration and Code Management:**
   - Use a **private GitHub repository** for code storage and collaboration.
   - Enforce version control best practices like branching and peer-reviewed pull requests.
4. **Security and Compliance:**
   - Ensure no organization-specific or sensitive data is uploaded to the repository.
   - Educate teams about **organizational security policies** to prevent violations.
5. **Evaluation:**
   - Conduct an **interim evaluation** to review individual progress and module-level completion.
   - The final evaluation assesses the fully integrated system for functionality, adherence to design principles, and team collaboration.
6. **Progress Tracking:**
   - Batch Admins monitor development progress using a **tracker**.
   - Include fields for module status, blockers, and milestones to ensure accountability.

Implementation Guidelines

1. **Team Formation and Role Assignment:**
   - Teams (PoDs) are formed with 5 members.
   - Assign one module per member, and appoint a **PoD Leader** for coordination and communication.
2. **Module Development:**
   - Each module adheres to **MVC architecture** for separation of concerns and scalability.
   - Developers implement both:
     - **Frontend**: User interface and interaction.
     - **Backend**: Business logic, database integration, and APIs.
3. **Agile Workflow:**
   - Conduct **sprint planning** to divide tasks into smaller, manageable units.
   - Hold **daily stand-ups** to discuss progress and blockers.
   - Conclude each sprint with a **sprint review** and retrospective.
4. **GitHub Repository Management:**
   - The **PoD Leader** creates and configures the private GitHub repository.
   - Use branching strategies such as main, feature/module-name, and hotfix.
   - Implement peer review for pull requests to ensure code quality.
5. **Collaboration Tools:**
   - Use **Microsoft Teams** for team communication.
   - Schedule regular sync-ups with the **Technical Trainer**, **Cohort Mentor**, and **Batch Admin** for guidance.
6. **Development Milestones:**
   - **Interim Milestones**:
     - Initial module design and implementation of basic features.
   - **Final Milestones**:
     - Complete system integration and end-to-end testing.
7. **Evaluation Process:**

- **Interim Evaluation**: Focuses on individual module design, code quality, and basic functionality.
- **Final Evaluation**: Assesses the integrated system, overall functionality, and team collaboration.

8. **Security Compliance:**
    - Repositories must not contain any **sensitive or organizational-specific data**.
    - Violations should be reported and resolved promptly.

9. **Tracking and Monitoring:**
    - Batch Admins maintain a detailed tracker with fields like module progress, blockers, and completion status.
    - Progress is reviewed at regular intervals to identify gaps early.

10. **Testing and Integration:**
    - Perform **unit testing** during module development.
    - Conduct **integration testing** to validate inter-module interactions.
    - Address edge cases and functional gaps during the testing phase.