Hands-On Lab Exercise for Module 4: Implementing Security and Compliance in GitHub

By the end of this exercise, participants will have practical experience in enhancing a repository's security and compliance posture, ready to apply these skills to their federal projects. They'll understand the importance of private repositories, branch protection, role-based access, security scanning, and maintaining a clear audit trail for compliance purposes.

Timeframe:

This exercise is designed to take approximately 30 minutes, providing hands-on experience with GitHub's security and compliance features tailored to federal contracting needs.

Scenario:

Upgrading Security Measures for a Federal Project Repository

Imagine you are leading a team on a federal project related to public health data management. The project involves handling sensitive health records and requires strict adherence to security and compliance standards. Your task is to upgrade the existing GitHub repository's security and compliance measures to meet these stringent requirements.

Starting Point:

1. **Access Your GitHub Account:**
   - Visit GitHub and log in with your credentials.
   - Navigate to the forked "GitHubFederalContractingCourse" repository.
2. **Navigate to the Module 4 Folder:**
   - Inside the repository, locate and click on the 'Module_4' folder visible in the file list.

Step-by-Step Lab Exercise Instructions:

1. **Setting Up a Private Repository:**
   - Navigate to the 'Settings' tab of your repository.
   - In the 'General' section, change the repository visibility to 'Private'.
   - Confirm the changes by following the prompts.
2. **Implementing Branch Protection Rules:**
   - Go to the 'Branches' section under 'Settings'.
   - Click on 'Add rule' for the main branch.
   - Set the rule to require pull request reviews before merging and include status checks you find necessary.
   - Enable 'Require review from Code Owners' and 'Include administrators'.
   - Save the changes to enforce these rules on the main branch.
3. **Managing Access and Permissions:**
   - Stay in the 'Settings' tab and navigate to 'Manage access'.
   - Click on 'Invite teams or people' to add collaborators.
   - Assign roles accordingly, granting 'Read' access to stakeholders and 'Write' or 'Admin' access to key team members.
   - Discuss the importance of regular audits and revoking access when no longer needed.
4. **Enabling Security Features:**
   - Navigate to the 'Security' tab on the repository.
   - Enable 'Dependabot alerts' and 'Dependabot security updates'.
   - Set up 'Code scanning alerts' by clicking on the 'Set up code scanning' button and following the instructions to enable and schedule regular scans.
5. **Auditing Activities:**
   - Still under the 'Security' tab, explore the 'Audit log' feature.
   - Review recent activities in your repository, filtering by date range or event type.
   - Discuss how understanding and monitoring these logs are crucial for maintaining security and compliance.
6. **Applying Compliance Documentation:**
   - Create a new file in the repository named 'Compliance.md'.
   - Document the security measures and compliance standards the project adheres to, referencing specific regulations and how the repository settings align with these requirements.
   - Commit the file to the main branch, ensuring it's visible for audit purposes.

Conclusion and Review:

Wrap up the exercise by reviewing the changes made to the repository. Ensure that the repository is now private, branch protection rules are in place, collaborators have appropriate access, security features are active, and compliance documentation is present and updated.

Reflect on how each action contributes to the overall security and compliance of the project. Consider the ongoing responsibilities of monitoring, updating, and documenting to maintain these standards.