CSP 451 -CheckPoint 1

Lab Assignment: Introduction to Version Control with Git and GitHub

Lab Objective:
The purpose of this lab is to help students understand the fundamentals of version control, learn the basic features of Git, and utilize GitHub for collaborative development and project management.

Lab Overview

By the end of this lab, students will:

1. Understand the concept and purpose of version control.
2. Learn basic Git commands and operations such as initializing repositories, committing changes, and creating branches.
3. Create and manage a GitHub repository.
4. Write a well-structured README file using Markdown.
5. Collaborate using GitHub features such as pull requests and issue tracking.

Lab Instructions

Part 1: Understanding Version Control

1. In your own words, explain the importance of version control in software development. Include:
    - How it helps track changes.
    - Collaboration benefits.
    - Backup and recovery mechanisms.
2. Submit a document with your explanation.

Part 2: Setting Up Git

1. Install Git on your local machine (if not already installed).
   - Use this guide for assistance.
2. Configure Git with your name and email:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Part 3: Creating a GitHub Repository

1. Log in to GitHub and create a new **private repository** with the following details:
   - Repository Name: `intro-to-git-lab`
   - Description: "This repository is for the version control lab assignment."
   - Initialize the repository with a README file.

Part 4: Basic Git Operations

1. Clone the repository to your local machine:

```
git clone <repository-URL>
```

2. Create a new file called `version-control-basics.md` in your local repository. Add the following content:
   - Definition of version control.
   - Key features of Git (refer to the material provided).
   - Save the file.
3. Stage and commit the new file:

```
git add version-control-basics.md
git commit -m "Add version control basics file"
```

4. Push the changes to the remote repository:

```
git push origin main
```

Part 5: Branching and Merging

1. Create a new branch called `feature-branch`:

```
git checkout -b feature-branch
```

2. Edit the `README.md` file to include your name and a short description of your favorite programming language.
3. Stage, commit, and push the changes to the `feature-branch`.

```
git add README.md
git commit -m "Update README with my name and favorite pro
gramming language"
git push origin feature-branch
```

4. Create a pull request on GitHub to merge `feature-branch` into the `main` branch.
5. Add a classmate or the instructor as a reviewer.
6. Once the pull request is approved, merge it into the `main` branch.

Part 6: Writing Documentation Using Markdown

1. Update the `README.md` file in your repository with the following sections:
   - **Project Overview:** A brief description of the lab assignment.
   - **Key Concepts:** Summarize the importance of version control and Git features.
   - **How to Run This Project:** Instructions on cloning the repository and viewing the content.
2. Format the README using Markdown syntax:

- Use headings, bullet points, links, and any other appropriate elements.

Part 7: Collaboration

1. Invite a classmate to your repository as a collaborator.
2. Assign them an issue to add a new file to the repository describing their favorite Git command.
3. Monitor their progress using the GitHub issues tracker and pull requests.

Lab Submission

Submit the following by the end of the lab session:

1. A link to your GitHub repository.
2. A PDF export of your repository's README file.
3. A short report detailing the steps you followed and any challenges faced.

Bonus Task (Optional)

Create a Markdown table in your README file summarizing key Git commands, their syntax, and functionality. For example:

Command	Syntax	Description
`git clone`	`git clone <URL>`	Clone a remote repository to local.
`git commit`	`git commit -m ""`	Commit changes with a message.

Grading Criteria:

- Completeness of the tasks.
- Correct usage of Git and GitHub features.
- Quality and formatting of the README file.