Below is the updated version of the GitHub repository structure with tasks that students can solve for each section (Version Control and Issue Tracking). I’ve added tasks to encourage hands-on practice for students.

### 1. \*\*Create the GitHub Repository\*\*

- Follow the initial steps to create a GitHub repository as described earlier.

---

### 2. \*\*Folder Structure\*\*

The folder structure will remain similar to the previous example but now includes \*\*Task Solutions\*\* and additional task descriptions for students to solve.

```

.

├── README.md

├── VersionControl

│ ├── Lab Instructions.md

│ ├── Tasks

│ │ ├── Task1.md

│ │ ├── Task2.md

│ │ ├── Task3.md

│ └── Solutions

│ ├── Task1-Solution.md

│ ├── Task2-Solution.md

│ ├── Task3-Solution.md

├── IssueTracking

│ ├── Lab Instructions.md

│ ├── Tasks

│ │ ├── Task1.md

│ │ ├── Task2.md

│ └── Solutions

│ ├── Task1-Solution.md

│ ├── Task2-Solution.md

└── Resources

└── references.md

```

---

### 3. \*\*Version Control Section (`02-Version Control`)\*\*

\*\*Folder\*\*: `VersionControl/Lab Instructions.md`

```markdown

# Lab 2: Version Control

## Objectives:

- Learn Git commands and workflows to manage project versions.

- Perform common operations: clone, commit, push, pull, branch, and merge.

- Practice resolving merge conflicts in a collaborative environment.

## Overview:

Complete the tasks in this lab to gain hands-on experience with Git. Each task is followed by steps that students should complete, along with challenges to solve independently. Make sure to follow each task in order.

### Lab Tasks:

1. \*\*Task 1: Cloning a Repository\*\* - Follow along with the steps and practice cloning a repository.

2. \*\*Task 2: Making Changes and Committing\*\* - Edit files, commit changes, and push them to the repository.

3. \*\*Task 3: Branching, Merging, and Resolving Conflicts\*\* - Explore working with branches and resolving merge conflicts.

Each task includes instructions, but you will also find a \*\*Student Task\*\* section where you are expected to complete an additional challenge to test your understanding.

```

\*\*Folder\*\*: `VersionControl/Tasks/Task1.md`

```markdown

# Task 1: Cloning a Repository

## Goal:

Clone an existing GitHub repository and prepare it for local development.

## Instructions:

1. Open your terminal (or Git Bash).

2. Clone the repository:

```bash

git clone https://github.com/yourusername/CMPS310-VersionControl-IssueTracking-Lab.git

```

3. Navigate into the cloned directory:

```bash

cd CMPS310-VersionControl-IssueTracking-Lab

```

4. Confirm the repository has been cloned by listing its contents.

### Student Task:

- Clone another public repository of your choice from GitHub.

- List the repository files using Git commands.

- Identify the default branch of the repository.

### Expected Output:

- List of repository files in your terminal.

- Confirmation of default branch.

```

\*\*Folder\*\*: `VersionControl/Tasks/Task2.md`

```markdown

# Task 2: Making Changes and Committing

## Goal:

Make changes to a file and push them to the remote repository.

## Instructions:

1. Create a new file called `student.txt` and add some content:

```bash

echo "Student Name: [Your Name]" > student.txt

```

2. Add the new file to the staging area:

```bash

git add student.txt

```

3. Commit the changes:

```bash

git commit -m "Added student.txt with my name"

```

4. Push the changes to the remote repository:

```bash

git push origin main

```

5. Verify that the changes are reflected in the GitHub repository.

### Student Task:

- Create a second file, `coursework.txt`, and add your favorite course names to it.

- Commit the file and push it to the repository.

- Review your commit history using:

```bash

git log

```

### Expected Output:

- The new `student.txt` and `coursework.txt` files should be present in the GitHub repository.

- A list of recent commits in your terminal.

```

\*\*Folder\*\*: `VersionControl/Tasks/Task3.md`

```markdown

# Task 3: Branching, Merging, and Resolving Conflicts

## Goal:

Learn to create branches, make changes, and resolve conflicts.

## Instructions:

1. Create a new branch for adding a new feature:

```bash

git checkout -b add-feature

```

2. Modify the `README.md` file by adding a new section, "Future Features".

3. Add and commit the changes in the new branch.

4. Switch back to the `main` branch:

```bash

git checkout main

```

5. Merge the `add-feature` branch into `main`:

```bash

git merge add-feature

```

6. If there are conflicts, edit the conflicting files, resolve the issues, and commit the changes.

### Student Task:

- Create another branch called `bug-fix` and simulate a small bug fix by modifying any file.

- Make sure that the `README.md` file is edited in both the `bug-fix` and `add-feature` branches to simulate a merge conflict.

- Merge the branches and resolve the conflict.

### Expected Output:

- Changes should be merged successfully into the `main` branch, and conflicts resolved manually if they occur.

```

---

### 4. \*\*Issue Tracking Section (`03-Issue Tracking`)\*\*

\*\*Folder\*\*: `IssueTracking/Lab Instructions.md`

```markdown

# Lab 3: Issue Tracking

## Objectives:

- Learn how to track and manage issues using GitHub.

- Create, assign, and close issues.

- Use labels and milestones to organize issues effectively.

## Overview:

In this lab, you will practice managing issues for a GitHub project. The lab involves tasks like creating, assigning, and resolving issues. Each task will have a challenge for students to solve.

### Lab Tasks:

1. \*\*Task 1: Creating an Issue\*\* - Create and label an issue.

2. \*\*Task 2: Assigning and Closing Issues\*\* - Work on resolving and closing issues.

Each task will also include a \*\*Student Task\*\* section where you need to manage issues beyond the examples.

```

\*\*Folder\*\*: `IssueTracking/Tasks/Task1.md`

```markdown

# Task 1: Creating an Issue

## Goal:

Create a new issue on GitHub for a feature or bug.

## Instructions:

1. Go to the \*\*Issues\*\* tab of your GitHub repository.

2. Create a new issue titled "Improve Documentation" with a description of adding more detail to the `README.md` file.

3. Add a label like `enhancement`.

4. Assign the issue to yourself.

### Student Task:

- Create a second issue titled "Fix bug in login functionality" with the `bug` label.

- Add a detailed description of the bug (e.g., "Login fails when entering invalid credentials").

- Assign the issue to one of your teammates (or yourself if working individually).

### Expected Outcome:

- The newly created issues should be listed under the \*\*Issues\*\* tab with appropriate labels and assignments.

```

\*\*Folder\*\*: `IssueTracking/Tasks/Task2.md`

```markdown

# Task 2: Assigning and Closing Issues

## Goal:

Assign and resolve an issue, and link it to a code change.

## Instructions:

1. Assign the "Improve Documentation" issue to yourself if it's not already assigned.

2. Make changes to the `README.md` file, for example by adding a "Getting Started" section.

3. Commit the changes and include a reference to the issue number in the commit message:

```bash

git commit -m "Improved documentation in README, fixes #1"

```

4. Push the changes to GitHub:

```bash

git push origin main

```

5. Verify that the issue is automatically closed.

### Student Task:

- Choose the second issue you created ("Fix bug in login functionality"), resolve it in your code, and follow the same process to commit and close the issue.

### Expected Outcome:

- The issues should be marked as \*\*Closed\*\* in the GitHub repository, with their respective commits linked to the issues.

```

---

### 5. \*\*Resources Folder\*\*

\*\*Folder\*\*: `Resources/references.md`

This folder remains the same and contains useful links for students to reference during their work.

```markdown

# References

1. [Git Documentation](https://git-scm.com/doc)

2. [GitHub Issue Tracking Guide](https://guides.github.com/features/issues/)

3. [Version Control with Git - Atlassian](https://www.atlassian.com/git/tutorials)

4. [Git Branching and Merging](https://www.git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging)

```

---

### 6. \*\*Solutions Folder\*\*

The \*\*Solutions\*\* folder for each task can contain solutions for instructors or as references for students after the lab.

---

This structure encourages students to solve tasks themselves, providing them with hands-on experience. The tasks are detailed enough to guide them through various Git and Git