

New chat

Today

Create GitLab Repo

Accessible Web Design Tips

Renew Plus

BI

bikas.neupane2008@g...

...

BI

Create a repository on gitlab.une.edu.au and push the main branch of this repository to it entirely unmodified. Please create the repository under your username, not under the cosc220-2022 group. You will need to set the visibility on your repository so that the marker can read from it (or, preferably, add the unit staff from Project information -> Project members). Repositories we cannot access cannot be marked (and will receive 0 until you can fix it)

Hint: Untick the option "Initialize this project with a README" when you create your project on gitlab.

Set your name and email on the repository for your future commits, using git config

Update the "name" label in the application to show your name. Commit this change and push it to your main branch.

Create an Issue in your GitLab project for fixing the errors in the assignment.

Create and check out a branch for your bugfix. This should follow the convention of having the issue number, followed by a hyphenated description of the issue. e.g. 1-fix-assignment-errors.

Push this branch to your remote repository. (From here on, I'll refer to this as your "issue branch")

Create unit tests that will detect the errors in the code.

Commit these unit tests with an appropriately descriptive commit message.
Do not fix the bugs in this commit. The unit tests should fail.
Push this commit to your remote repository (on your issue branch)

Tag this commit, with the label testfail. Push this tag to your remote repository.


Fix the bugs in the code. Run your unit tests again. The tests should now pass.

Commit your bugfix. Push it to your issue branch.

Check out main. Merge your issue branch to main. The tests should still pass. Push this to your gitlab repository.

Close your Issue with a reference to the commit hash in which you fixed the bugs.

Paste the link to your gitlab repository into the assignment (on Moodle, below) and submit.



I'm sorry, but as an AI language model, I don't have direct access to external systems or the ability to perform actions on specific platforms like GitLab. Therefore, I cannot create a repository on GitLab for you or perform the specific tasks you mentioned.

However, I can guide you through the process of creating a repository on GitLab and performing the mentioned tasks using Git commands. Here's a step-by-step guide:

- Create a repository on GitLab:
 - Go to [GitLab \(gitlab.une.edu.au\)](https://gitlab.une.edu.au) and log in with your account.
 - Click on the "New project" button to create a new repository.
 - Choose a name for your repository and provide an optional description.
 - Untick the option "Initialize this project with a README."
 - Set the visibility of the repository to a level where the marker can read from it or add the unit staff as project members.
- Set your name and email for future commits:
 - Open a terminal or Git Bash.
 - Navigate to the directory of the project you want to push to the repository.
 - Run the following commands, replacing "Your Name" and "[your@email.com](#)" with your actual information:

arduino

Copy code

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

3. Push the main branch to the repository:
 - In the terminal or Git Bash, navigate to the project directory.
 - Initialize a new Git repository if not already initialized: `git init`
 - Add all the files to the repository: `git add .`
 - Commit the changes: `git commit -m "Initial commit"`
 - Add the GitLab repository as a remote: `git remote add origin <repository_url>`
Replace `<repository_url>` with the URL of the GitLab repository you created.
 - Push the main branch: `git push origin main`
4. Update the "name" label in the application:
 - Make the necessary changes to update the "name" label in your application.
 - Commit the changes: `git commit -am "Update name label"`
 - Push the changes to the main branch: `git push origin main`
5. Create an issue in your GitLab project:
 - Go to your GitLab repository.
 - Click on the "Issues" tab.
 - Click on the "New issue" button.
 - Provide a title and description for the issue, highlighting the assignment errors.
6. Create and check out an Issue branch:
 - In the terminal or Git Bash, navigate to the project directory.
 - Create a new branch following the issue branch naming convention: `git checkout -b 1-fix-assignment-errors`
Replace "1" with the issue number and "fix-assignment-errors" with a hyphenated description of the issue.
7. Create unit tests for error detection:
 - Write unit tests that detect the errors in the code.
 - Commit the unit tests: `git commit -am "Add unit tests for error detection"`
8. Push the issue branch and tag the commit:
 - Push the issue branch to the remote repository: `git push origin 1-fix-assignment-errors`
 - Tag the commit with "testsfail": `git tag testsfail`
 - Push the tag to the remote repository: `git push origin testsfail`
9. Fix the bugs in the code and run the unit tests:
 - Make the necessary fixes in your code.
 - Run the unit tests to ensure they pass.
10. Commit and push the bugfix to the issue branch:
 - Commit the bugfix: `git commit -am "Fix assignment bugs"`
 - Push the bugfix to the issue branch: `git push origin 1-fix-assignment-errors`
11. Merge the issue branch to main:
 - Switch to the main branch: `git checkout main`
 - Merge the issue branch: `git merge 1-fix-assignment-errors`
 - Push the changes to the main branch: `git push origin main`
12. Close the issue:
 - Go to your GitLab repository.
 - Click on the "Issues" tab.
 - Find the issue you created.
 - Add a comment referencing the commit hash where you fixed the bugs.
 - Close the issue.

Once you have completed these steps, you can copy the link to your GitLab repository and submit it for your assignment. Remember to provide the necessary permissions or add the unit staff as project members so they can access your repository.

🔄 Regenerate response

Send a message

