**TransitHub Project Documentation**

Version Control Workflow

Before making any changes to the project, it's essential to follow the version control workflow to ensure smooth collaboration and avoid conflicts. The following steps outline the recommended workflow for making changes to the TransitHub project:

**1.Clone the Repository:**

Start by cloning the TransitHub repository to your local machine using the following command:

**git clone https://github.com/Ratshili9/TransitHub.git**

1. **Switch to the Main Branch:**

Ensure that you're working on the latest version of the project by switching to the main branch:

**git checkout main**

3. **Pull Latest Changes:**

- Before making any changes, pull the latest changes from the remote repository to synchronize your local copy with the main branch:

**git pull origin main**

**4. Create a New Branch:**

- Create a new branch for the feature or bug fix you're working on:

**git checkout -b feature-name**

Replace `feature-name` with a descriptive name for your branch.

**5. Make Changes:**

- Make your desired changes to the project files within your branch.

**6. Commit Changes:**

- Once you've completed your changes, commit them to your local branch with a descriptive commit message:

**git add .**

**git commit -m "Description of changes"**

**7. Push Changes to Remote Repository:**

- Push your changes to the remote repository on GitHub:

**git push origin feature-name**

Replace `feature-name` with the name of your branch.

**8. Create Pull Request:**

- Finally, create a pull request on GitHub to merge your changes into the main branch:

- Navigate to the repository on GitHub.

- Click on the "Pull Requests" tab.

- Click on the "New pull request" button.

- Select your branch and the main branch for the base.

- Provide a title and description for your pull request.

- Click on the "Create pull request" button.

**9. Review and Merge:**

- Your team members will review your pull request, provide feedback if necessary, and merge it into the main branch once approved.

**10. Cleanup:**

- Once your changes are merged, you can delete the feature branch both locally and remotely:

**git checkout main**

**git branch -d feature-name**

**git push origin --delete feature-name**

By following this version control workflow, we can ensure that all changes are properly managed and integrated into

the project while minimizing conflicts and disruptions.

**Hope this docx shows up better on ur pc’s guys**