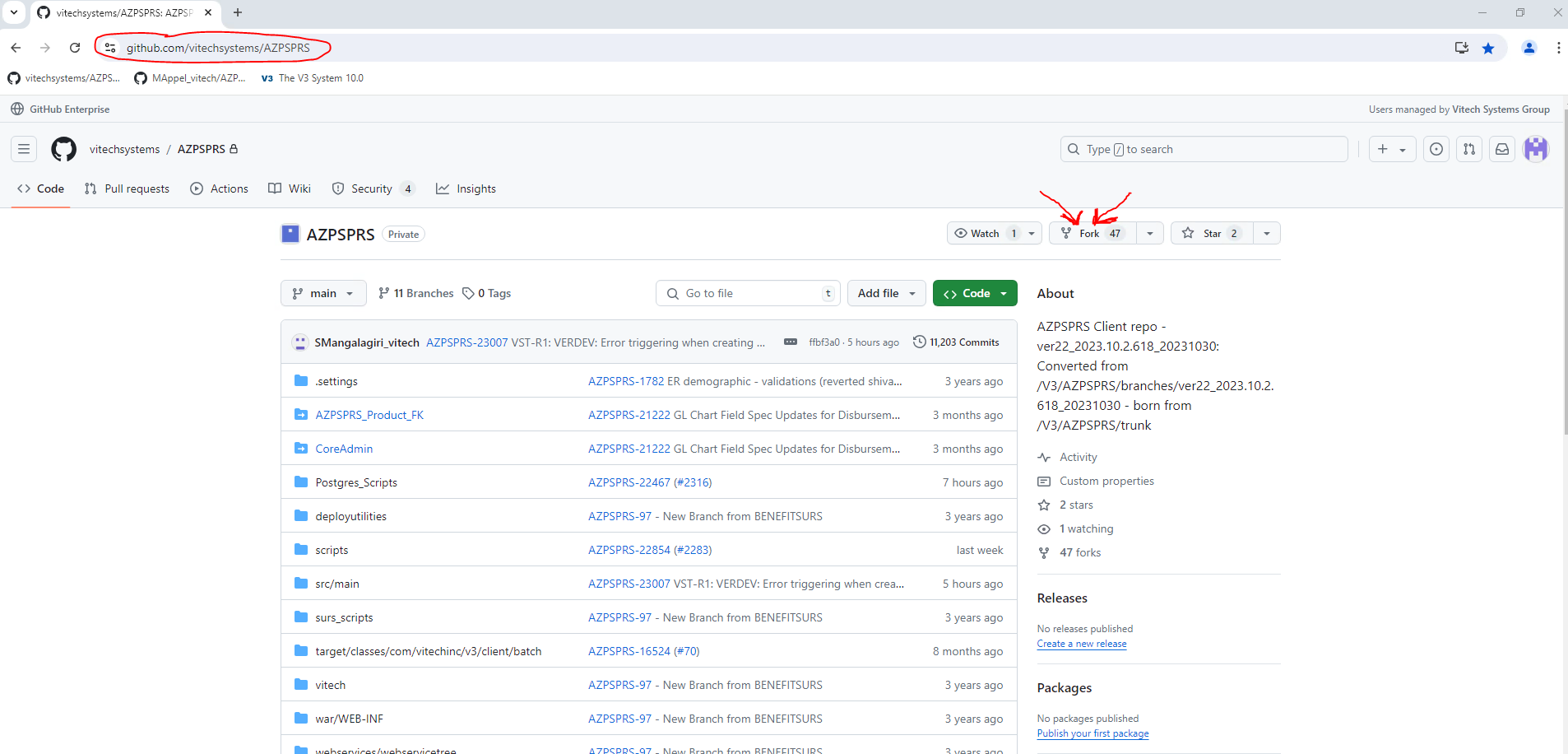
## **Section I - Source Control Management with git**

* GitHub Access

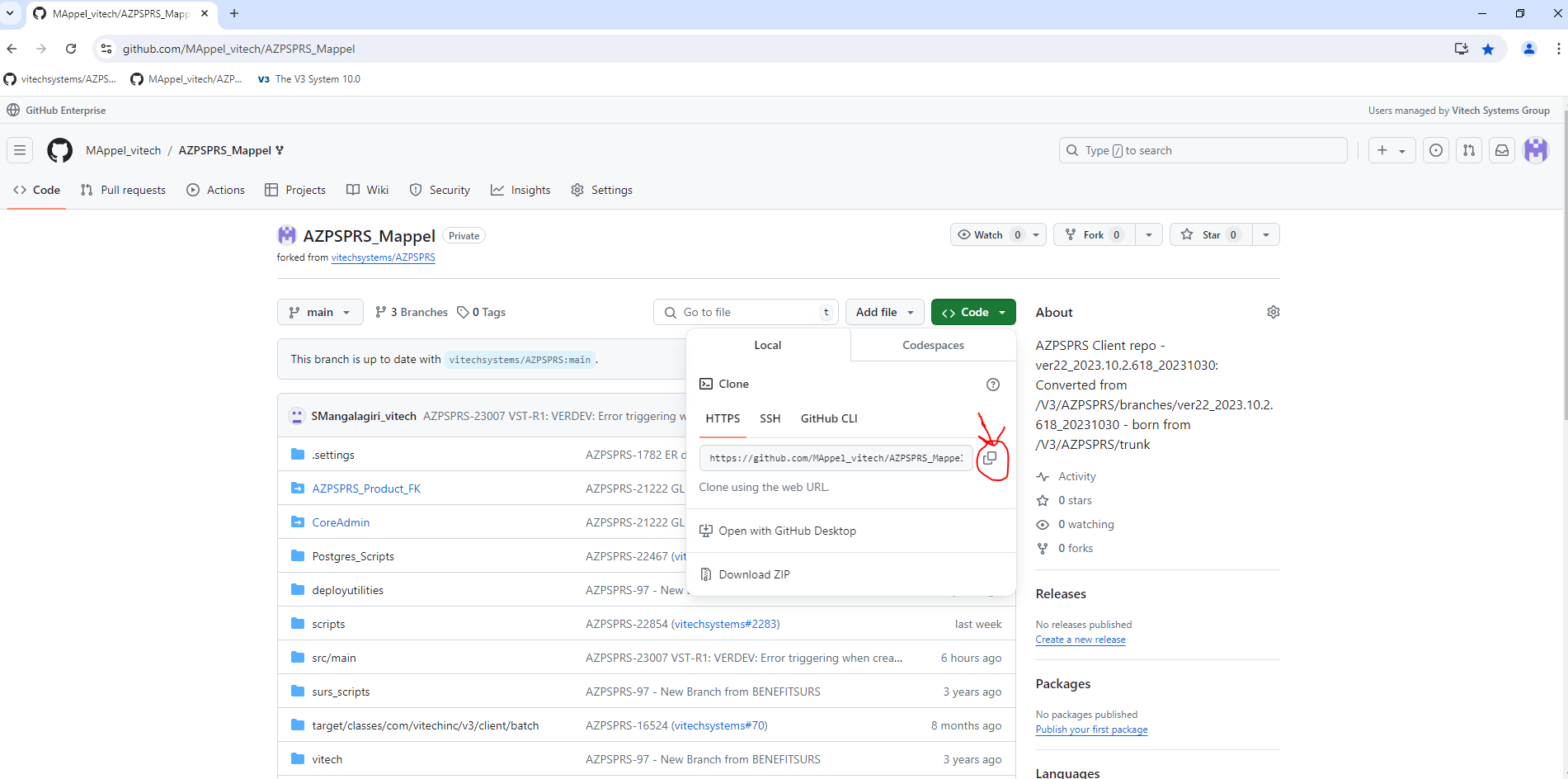
1. Verify your GitHub Access by opening the SSO url (<https://github.com/enterprises/vitech>) in chrome
2. If it does not recognize your username, use <username>\_vitech and then use the vitechb2c password to log on
3. Once you have verified access in chrome also make sure Chrome is your default browser, as we will use the browser login mode to authenticate to GitHub from the command prompt.

* Go to vitech AZPSPRS main github repository page

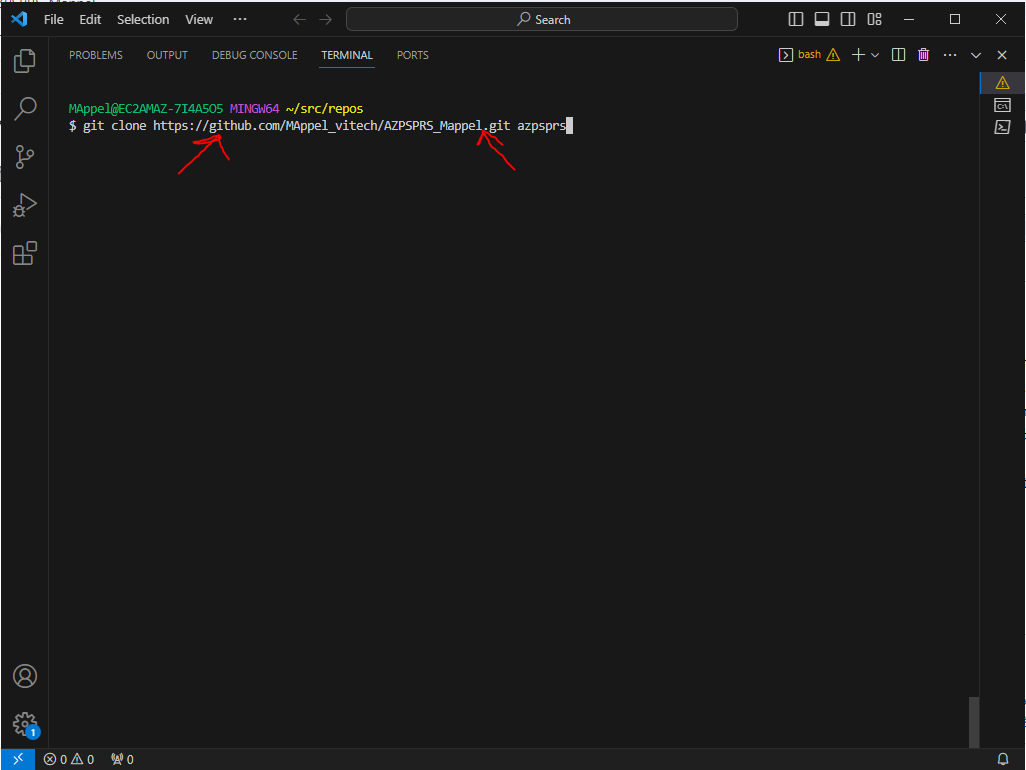
1. Click fork button to create a personal forked repository github page



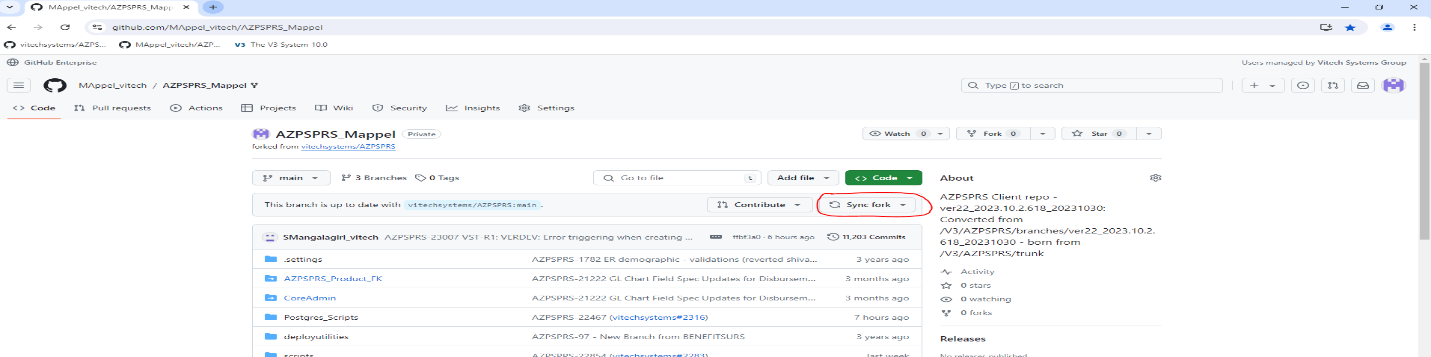
* Click green code button to expand and then click icon to copy clone url to clipboard

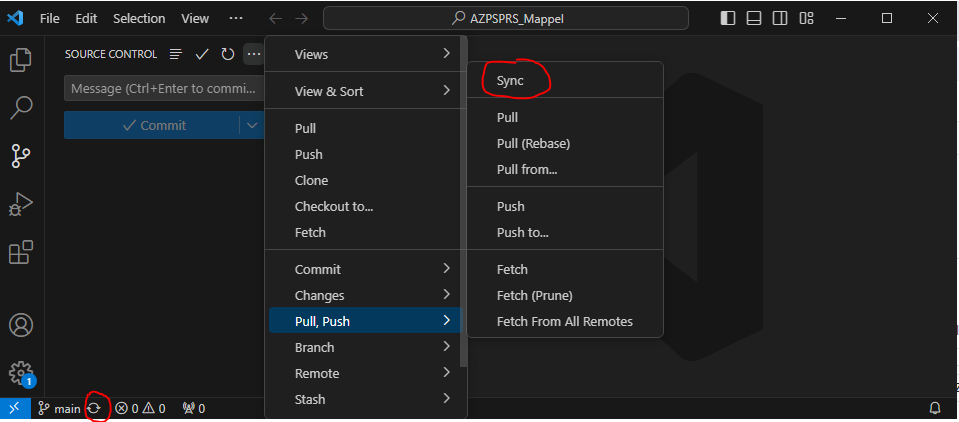


* Go to your local file system and clone your forked repository

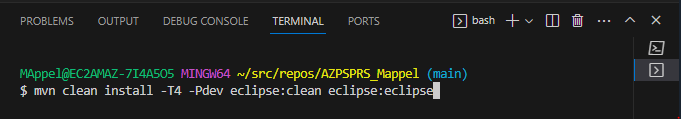


* Add the .gitignore file to root of repo. This file instructs Git which type of files not be considered for Git Tracking. (.gitignore file can be found [here](#_References))
* NOTE: Always make sure that your local is in sync with Forked Repo and Forked Repo in sync with Vitech Main UpStream.

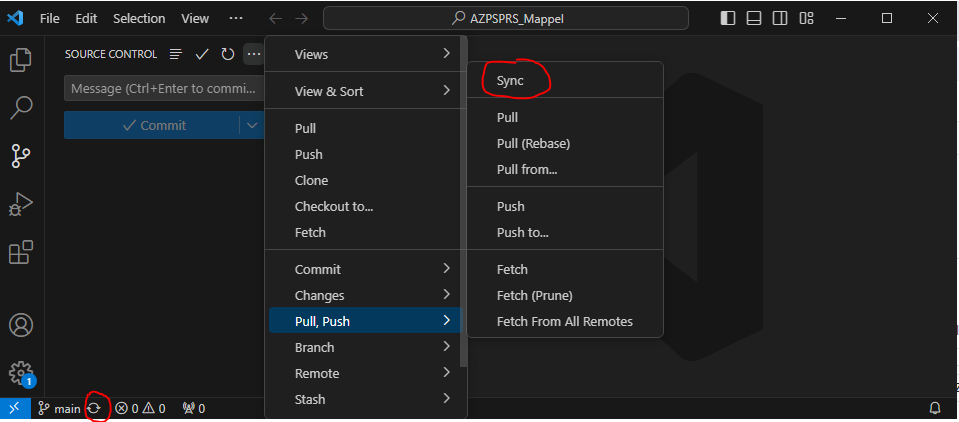




* Create feature branch from vscode gui or git bash (Check out these videos for vscode gui git commands - [vscode git basics 1](https://youtu.be/z5jZ9lrSpqk?list=PL30AETbxgR-fiCTXm4o-uDnngyIie3Wf5), [vscode git basics 2](https://youtu.be/Dedz4gRHezg?list=PL30AETbxgR-fiCTXm4o-uDnngyIie3Wf5) or check udemy for vscode and git training)
* Do clean build with command, use the proper given mvn command to build

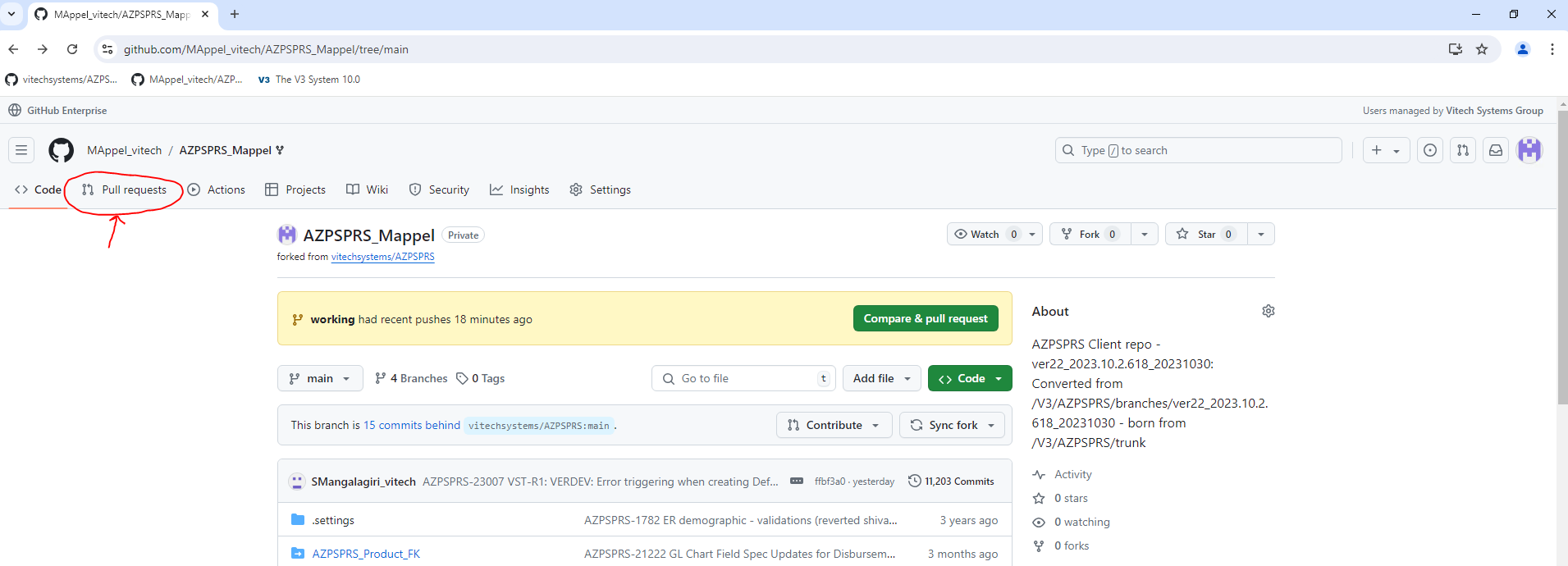


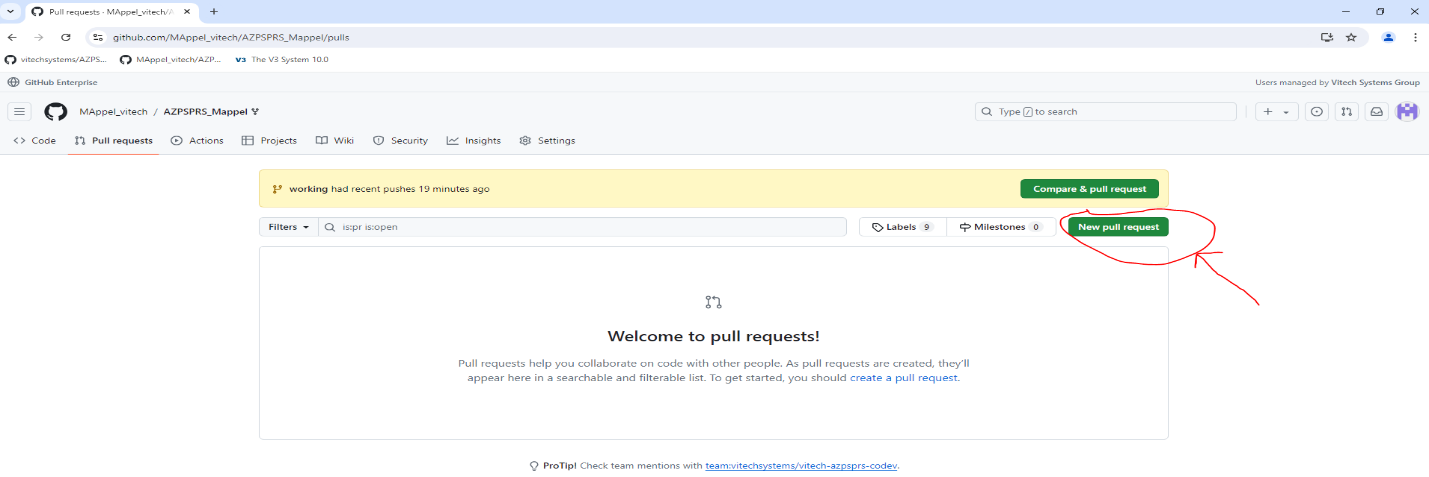
* Import the project into eclipse. Do code changes and unit testing. (See V3locity AZPSPRS CoDev doc for details on setting up eclipse for code dev and debug)
* From the vscode editor, open project folder then switch to source code view and commit changes to local with proper JIRA ticket number. Then push(Sync) your selected changes to Forked Repo.



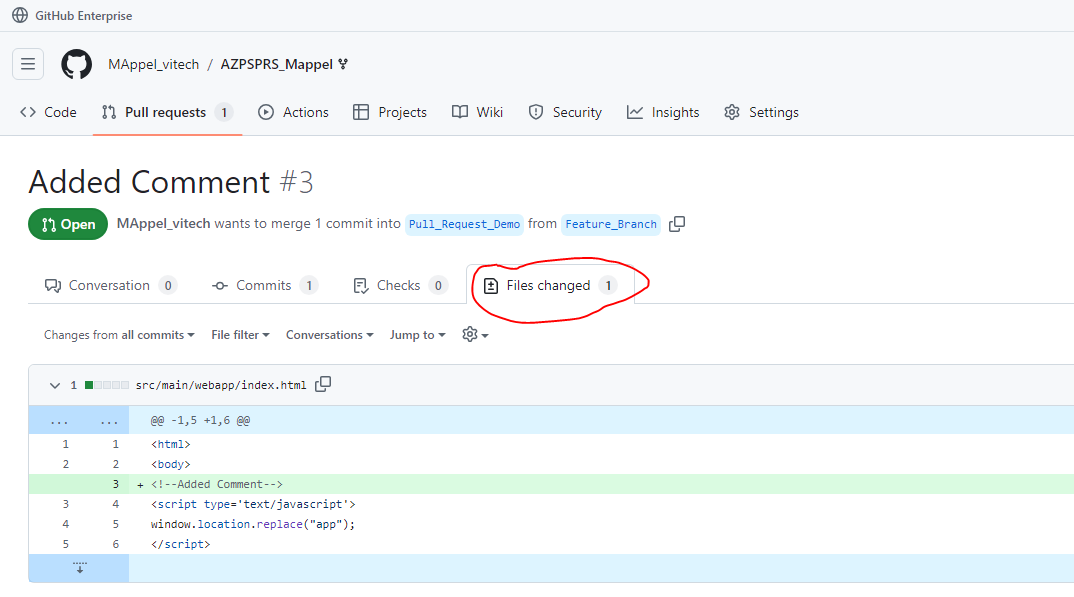
* Create pull request on forked server to merge main back into vitech azpsprs main. Add a PR comment referencing the Jira ticket with the following format :

*"This pull request relates to AZPSPRS-9876”  
 e.g.* ***AZPSPRS Master Jira:*** *AZPSPRS-9876 - "Release 1.0.0"*



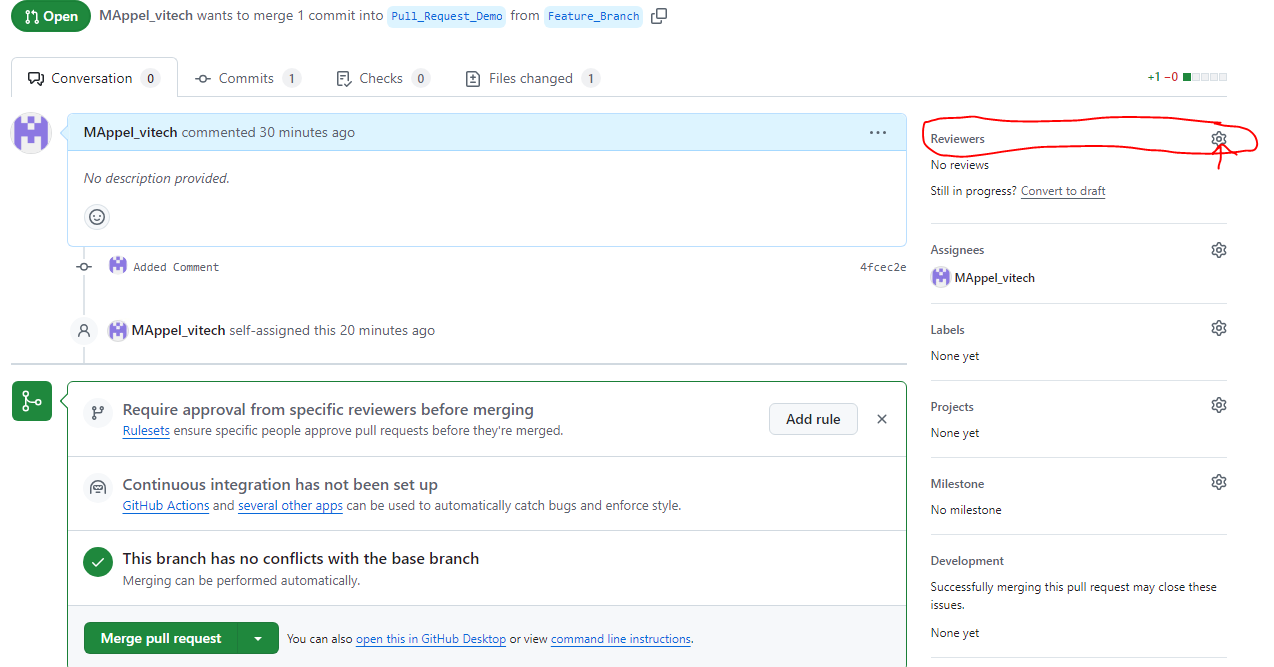


* Make sure "Files changed" contains only your changes before adding Reviewers. If not close the PR and re-create with your changes.

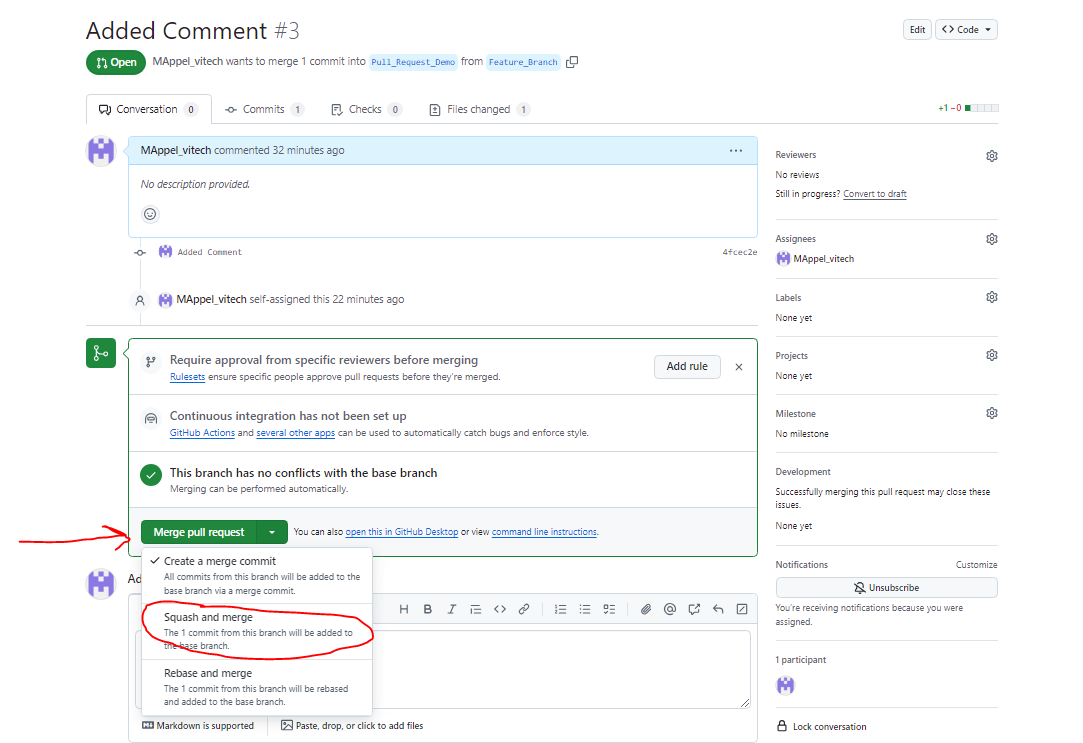


* Add Reviewers for the Pull Request

|  |  |  |
| --- | --- | --- |
| **Sr** | **Artifact** | **Reviewers** |
| 1 | Documents | Eswara Vayugundla  Subramanyam Nistala  Sridhar Kammary Sidharth Kamat |
| 2 | Query | Anil Atyam  Eswara Vayugundla  Venkatesh Racharla  Sridhar Kammary Sidharth Kamat |
| 3 | Reports | Ramesh Ausali  Nishanth Kona  Jagadeesh Kummara Sridhar Kammary Sidharth Kamat |
| 4 | Business Rule | Jagadeesh Kummara Suresh Mangalagiri Sridhar Kammary Sidharth Kamat |



* After reviewer(s) have approved the changes Squash and Merge changes to upstream (vitechsystems/AZPSPRS main branch)



## **Section II – Deployment Workflow**

* The information below from Vitech is an example of the overall deployment workflow, including any required documentation and testing procedures. Use the steps Section I of this document as a reference to implement the workflow.

Most fixes will consist of either Java Files and SQL scripts or both.

* All java and sql files are to be committed to github (as per provided training)
* All sql files must be placed in the script flush queue detailed below.

Once the files have been placed in the queue and github, the next scheduled build will do the following *(currently builds are scheduled to run daily @ 10:30 AM EST, followed by deployments to all test environments)*:

1. Build a war file with all committed java changes
2. Deployments to all environments in the promotion path. A Deployment consists of
   1. Executing/ Flushing of all sql scripts
   2. Deploying of the war file generated in #1.

**Process to place files in the queue:**

You must drop your changed/written SQL scripts in the below location always. If their SQL script-1 is dependent on SQL Script-2, then place the script-2 first and then script-1 or else if you drag and drop all files, system will allow you to sort files in the right order in a pop-up window. Once the script files are dropped, V3 automatically prepends timestamp to each script file.

The bamboo CI/CD pipeline will use these timestamps to determine the order to flush scripts.

AZPSPRS TRUNK Q site: <https://scripts.vitechinc.com/s3list.php?dir=AZPSPRS/PG_TRUNK_QUEUE/_(OPEN-CLIENT-SCRIPTS-ONLY)/>

**Example:**

For example, if a report is being developed. For this you must prepare a script to register an entry in report\_definition, Business\_rule, business\_rule\_expr, parameter\_group and parameter\_group\_detail table for example ticket AZPSPRS-100 / EmployerReport.

Once the script is ready. Let’s azpsprs-100-employer\_report-1.sql

1. Run the script azpsprs-100-employer\_report-1 in AZPSPRSDEV Instance

2. Develop the Callback java file (For Rendering Models, defaulting values, validating field values), Let’s say Java file Name: EmployerReportTest.java and bean registered in employer\_report\_config.xml

3. Design the report design and the final file is: employer-report.jrxml

4. Integrate all and test, and if all looks good. Please follow below steps:

1. Create Pull Request (PR) for all above files:
2. azpsprs-100-employer\_report-1.sql
3. EmployerReportTest.java
4. Employer\_report\_config.xml
5. employer-report.jrxml
6. Attach the Unit test document to the PR
7. Add the right reviewers for type of issues/ fix as Reviewers and create Pull request.
8. Change the Ticket status from “In Development” to “Code Review” and add the PR (Pull Request URL) detail ticket step details and send it to Sidharth Kamat.
9. Once file changes look good, we will either Fail the PR or approve the PR changes. If all looks good, we will add comments in PR and in step notes to commit changes. Then Go to Step#vii)
10. If failed, close the existing PR and make changes at required places, again process starts from STEP# i)
11. Once we approve, you can ‘Merge’ changes to AZPSPRS Main Repo.
12. Place the SCRIPT-1.sql file in AZPSPRS Queue Flush folder (If you place the script in Queue flush folder, then only configuration(script) changes will be propagated to sub-sequent environments)

A screenshot of a computer

Description automatically generated

## **Refere****nces**

