**Development Environment Setup Guide**

**Environment Software/Tools Setup:**

1. Install Python 3.11.1
   1. <https://www.python.org/downloads/release/python-380/>
2. Open command prompt and check the Python version.
   1. Run command “python –version” to check the version of Python.
   2. If the command is not recognized, set the environment path for Python and run “python –version” command again.
3. Install Pip environment using the command below.
   1. pip install pipenv
   2. If the command is not recognized, set the environment path for Pip and run the command again.

**Code Setup:**

1. Clone this git repository or download the zip from git
   1. Git: <https://github.com/se691-group3/proof-buddy>
2. Open command prompt and navigate to the top folder of application.
3. Run command “pipenv install” to install all project dependencies and frameworks.
4. Run command “pipenv shell” to activate the virtual environment.
5. Run command “python manage.py makemigrations” to create migration files.
6. Run command “python manage.py migrate” to migrate data models to the SQLite database.
7. Run command “python manage.py collectstatic” to collect all objects in static folder.
8. Run command “python manage.py runserver” to initiate the server.
9. Go to any browser and enter below url to access the local environment.
   1. http://127.0.0.1:8000/

**Regression Testing Setup:**

1. Create a new file “rulename.py” file under proofchecker/rules.
2. Edit file “proofchecker/rules/rulechecker.py” and add proofchecker.rules.dubnegintro import DubNegIntro
3. Open command prompt and navigate to the top folder of application.
4. Run command “pip install -r requirements.txt” to setup all test packages.
5. Run command “python manage.py test” to execute the test cases.

**Git Branching Strategy:**

Git branches are used as a means for teams to develop features giving them a separate workspace for their code. These branches are usually merged back to a master branch upon completion of work. In this way, features (and any bug and bug fixes) are kept apart from each other allowing you to fix mistakes more easily. A branching strategy is the process development team adopts when writing, merging and deploying code when using a version control system.

For Proof Buddy project, we will follow the strategy with below branches:

* Master – This is the branch which will be deployed to production.
* Develop – This is the branch from where all feature branches will be created.
* Feature – This is the branch where all the features and code changes will be implemented. Once code changes are fully validated, they will be merged back to the develop branch.

Note: The Master and Develop branch will stay throughout the life of the project, however feature branched can be deleted once the code is merged in Develop branch.

A diagram of a process

Description automatically generated with low confidence

**Commands:**

**Git clone command:**

* git clone <remote origin="" url=""></remote>

**Creating a feature branch:**

* git checkout develop
* git checkout -b feature\_branch

**Merging a feature branch:**

* Finishing git checkout develop
* git merge feature\_branch