Google Summer of Code @ OSGeo

GSoC Proposal

1. Contact details

- Name and surname: Supreet Singh Gill
- Nickname: Sharry
- Country: India
- Email: sharrygill5656@gmail.com
- Phone: +919855750732
- Public repository/ies: Link to repository
- Personal blog: Link to blog
- Twitter/Identica/LinkedIn/others: Link to twitter

2. Project Idea

- OSGeo or guest software: GRASS GIS
- Title: Tool to create Automatic Module test writer in Python for GRASS GIS.
- Brief description of the idea :

The Tool will be completely automatic and allows the user to create test very easily.

1) First, there is an option of entering the name of module .

2)Then the user will press “Return or ENTER” and then a python file will be provided to the user which contains the test.

The user only have to enter the name of the module and the whole things in test(Flags, tests) will be automatic, here automatic means that the flags and tests will be already in the code according the requirements of the module . i.e. Everything will be in background, the user will only need to enter the name of module, and the flags and tests(rasterMinMax, SimpleModule, etc.) will be defined in the code according to the module. We can use object orientation, if most of the modules will have same parameters. Or we can define classes according to the modules.

There will be a drop down list of modules when the user will start entering the name, the drop down list will appear according the the alphabets entered by the user, for eg: if user enters “v”, it will display “v.buffer, v.vect.stats, v.profile,etc”.

If we use the script in the GRASS command shell, the user will have to type “test r.mode” and press enter, the test will be created, and when the user will start entering the modules name the drop down list will appear.

- The state of the software BEFORE your GSoC: Their is no such tool to create test automatically, but there is a framework for test which was also created by GSoC student.

- The addition that your project will bring to the software: With this tool users can also do tests which do not know unit testing in Python.

- Future developments: I will try to add all the modules in the tool and improve the tool. i.e. if in future new modules will be developed, I will add in the tool.

3. Timeline

3.1.Community Bonding Period (April 23 to May 13, 2017):

1. Write some test for modules in python
2. Study the code and the test framework
3. Discuss the functioning of the tool with mentors.

3.2.Official Coding Period (May 14 to August 14, 2018):

Official Coding Period Phase 1 (May 14 to June 10, 2018):

Week 1 (May 14 to May 20, 2018):	Define the functioning of tool, i.e. In the code, I will define functions of the tool, e.g. adding module names, drop down list of modules, etc.
Week 2 (May 21 to May 27, 2018):	Add all vector modules in the tool. i.e. I will define vector modules in code, so that their name can be entered and the tests for vector modules can be generated.
Week 3 (May 28 to June 3, 2018):	Write a brief documentation about, how the tool work.
Week 4 (June 4 to June 10, 2018):	1.To test the working of tool, i.e. tool is working properly or not.

Deliverable #1: I will deliver working tool with all the vector modules and documentation. i.e. **Complete working code of the tool in which all the vector modules** will be defined and if we enter the name for vector modules, the test will be out.

First evaluation period (June 11 to June 15, 2018):

1. Mentors evaluation time.

Official Coding Period Phase 2 (June 11 to July 8, 2018):

Week 5 (June 11 to June 17, 2018):	Add all the raster modules in the tool. i.e. I will define raster modules in code, so that their name can be entered and the tests for raster modules can be generated.
Week 6 to 8 (June 18 to July 8, 2018):	Create Documentation. i.e I will update the documentation with completed explanation about the tool.

Deliverable #2: I will deliver working tool with all raster and vector modules and documentation. I.e. **Complete working code of the tool in which all the raster and vector modules** will be defined and if we enter the name for raster and vector modules, the test will be out.

Second evaluation period (July 9 to July 13, 2018):

1. Mentors evaluation time.
2. Provide the completed working tool with vector and raster modules..

Official Coding Period Phase 3 (July 9 to August 5, 2018):

Week 9 to 10 (July 9 to July 22, 2018):	Add all the modules other than raster and vector(imagery, etc)
Week 11 (July 23 to July 29, 2018):	Fix bugs in the code.
Week 12 (July 30 to August 5, 2018):	Improve Documentation and beautify the code, so that code can be easily readable and tool can be studied with documentation.

Deliverable #3: I will deliver the tool with all the modules in it and all the unittest created for the code.

Final evaluation period (August 6 to August 14, 2018):

1. Wrap up my projects and submit final evaluation of my mentors.

4. Studies

- What is your School and degree?: I am in **Guru Nanak Dev Engineering College**, currently doing my bachelors in Civil Engineering.

- Would your application contribute to your ongoing studies/degree? If so, how?: Yes, because in Civil Engineering, a lot of maps are required. I need to create them by using some software so, GRASS GIS and many it's amazing modules will make my work easy.

5. Programming and GIS

- Computing experience: I use Ubuntu on daily basis. And I am good in Python and learning C++.
- GIS programming and other software programming: GRASS GIS
  Test created for r.mode module of GRASS GIS: Link To Test

6. GSoC participation

- Have you participated to GSoC before?: No
- Have you submitted/will you submit another proposal for this year's GSoC to a different org?: No

7. Time Availability

I will be available 40 hours / week, if needed can spend more. No restriction of time.

8.Why GRASS GIS?

As being a civil engineering student, I need to create many maps usually by hand, and GRASS GIS allows us to create limitless maps by using amazing different modules, so it makes easy for us to create different maps.

9.Sample Code for tool:

Link to Draft Script Through this code, the file will be created and the test will be written in the file, this code is very simple, in which I will write test for each and every module. If we want to be complex and beautiful we can use object orientation, and classes according to modules.