Spring2019 CS5551: Advanced Software Engineering

Department of Computer Science Electrical Engineering
University of Missouri - Kansas City

Version Control system for Deep Learning (DLV)

Team 6:
Sindhusha Tiyyagura (24)
Pradeepika Kolluru (12)
Sravan kumar Pagadala (21)
Dinesh kumar reddy Kusam (14)

GITHUB URL:
https://github.com/sindhusha-t/VCS-Git-for-deep-learning

VIDEO URL:
https://www.youtube.com/watch?v=p0Z6mslp40o

PPT URL:
https://drive.google.com/open?id=15k_qQV8vwhH7z8nJcK1RgbNCV03Ee_KC

ZENHUB URL:
https://github.com/sindhusha-t/VCS-Git-for-deep-learning#workspaces/vcs-for-deep-learning-5c60d48784111d4401fa452b/board?repos=169820599

I. INTRODUCTION:

DLV – Deep learning version control system – Web Application
Where users can register and login to their accounts

To view list of their projects,
To view different versions of experiments done in the model,
To easily view the changes done locally,
To easily commit particular experiment.

II. PROJECT GOAL:

MOTIVATION:

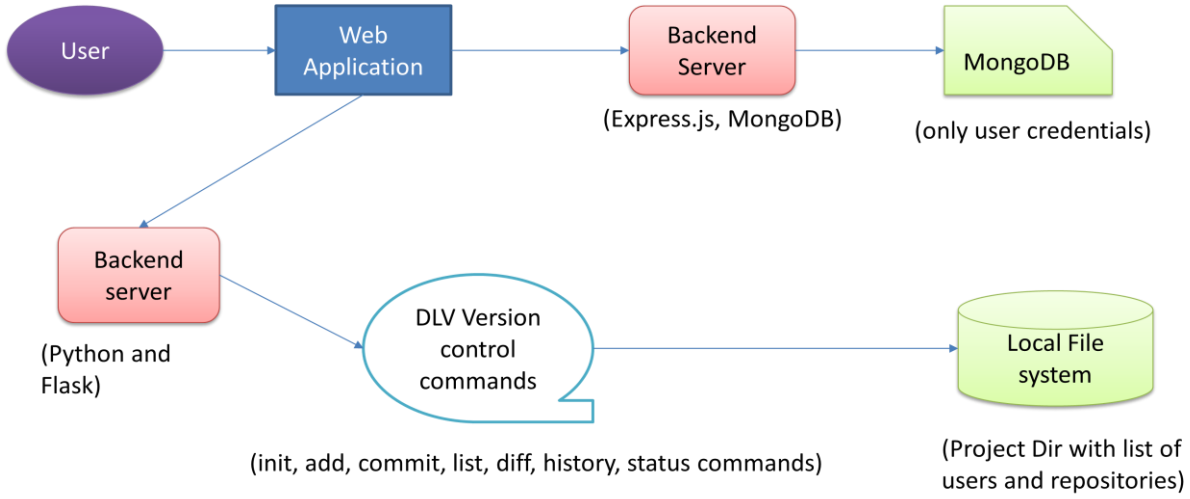
The main aim of the project is to develop a version control system for deep learning models (source code, input and output data files, metrics about the experiment) similar to GIT.

- There is no functionality in GIT to track group of files as a version (Deep learning is more dependent on the tracking of each experiment as a version)
- The following tasks can be done with the existing GIT functionalities like ***tags and branches***. But this is a very painful task where user has to remember that each experiment need to be tagged / given a branch.
- There is no functionality in GIT to compare the results of different experiments (this means the comparisons for different tags).

TECHNOLOGIES USED:

Angular
ExpressJS
MongoDB
NodeJS
Python and Flask
Heroku

SYSTEM ARCHITECTURE:



SYSTEM FEATURES:

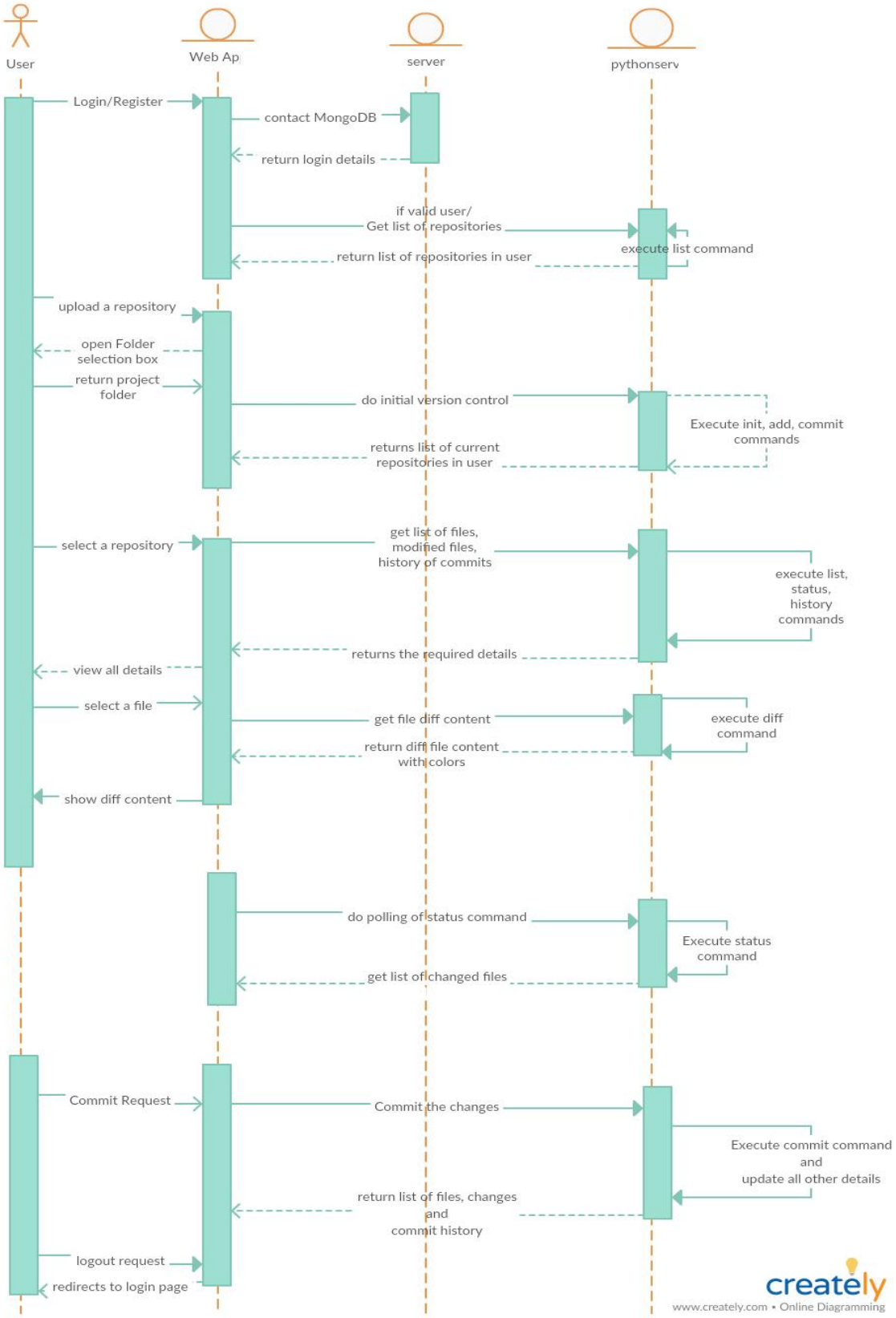
Command Line Interface for dlv commands:

- dlv init
- dlv add
- dlv commit
- dlv history
- dlv status
- dlv list
- dlv desc
- dlv diff
- dlv push

User Interface (Integrating the commands with UI):

- User Registration and Login
- User can view list of projects/deep learning models
- User can upload and start versioning the model.
- User can view the changes done locally in the system and verify the changes done before commiting the experiment.
- User can view history of all experiments and the changes in it.
- User can view the current version files/ last experiment that is saved.

SEQUENCE DIAGRAM:



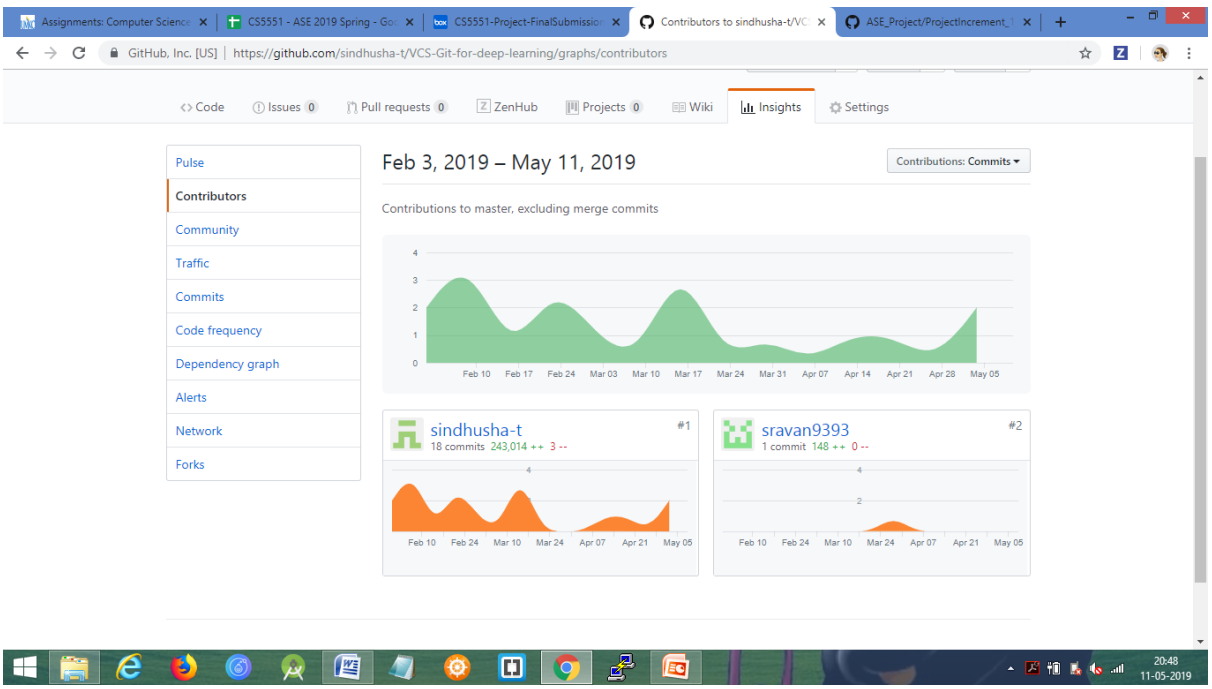
III. PROJECT PLAN:

The Screenshots of the project reports are attached here:

Burn down chart:

CONTRIBUTION REPORT:

Each one of us have done changes to the branches individually and then merged the changes with the master branch.



IV. PROJECT INCREMENT 1:

TASKS DONE:

1. Getting familiar with GIT open source code.
2. Understanding the requirements and coming up with the design for the functionality of dlv commands.
3. Implementing dlv init, add and commit commands.

DESIGN and IMPLEMENTATION:

Attaching the Screenshots of each command execution

1) DLV INIT:

Just like GIT – we followed the basic functionality like creating the .dlv directory which contains the information about versions and experiments of the models.

As we can see the folder structure in the screenshots where

- It has files like
    - HEAD.txt – points the current user branch
    - Config.txt – this contains the information about the project – like name, description, author/username
- It has folders like
    - Master – default branch,
        - Status.txt – contains the information about the changed files.
        - Commit_log.txt – contains the history of all commits/experiments.
        - Commits - this folder contains the at which stage the files are in for each experiment.
        - Cache – this folder contains the information of all the file versions. (It is created with the extensions like .1, .2, .3 ...)
    - For each user branch one folder similar to master is created

```
sindhusha55@instance-1:~/sample_project$ ls -lart
total 12
drwxrwxr-x  5 sindhusha55 sindhusha55 4096 Mar 18 15:58 Weather-Application
drwxr-xr-x 10 sindhusha55 sindhusha55 4096 Mar 19 02:32 ..
drwxrwxr-x  3 sindhusha55 sindhusha55 4096 Mar 19 03:53 .
sindhusha55@instance-1:~/sample_project$ dlv init
sindhusha55@instance-1:~/sample_project$ ls -lart
total 16
drwxrwxr-x  5 sindhusha55 sindhusha55 4096 Mar 18 15:58 Weather-Application
drwxr-xr-x 10 sindhusha55 sindhusha55 4096 Mar 19 02:32 ..
drwxrwxr-x  3 sindhusha55 sindhusha55 4096 Mar 19 03:54 .dlv
drwxrwxr-x  4 sindhusha55 sindhusha55 4096 Mar 19 03:54 .
sindhusha55@instance-1:~/sample_project$ ls -lart .dlv
total 16
drwxrwxr-x 5 sindhusha55 sindhusha55 4096 Mar 19 03:54 master
-rw-rw-r-- 1 sindhusha55 sindhusha55    6 Mar 19 03:54 HEAD.txt
-rw-rw-r-- 1 sindhusha55 sindhusha55    0 Mar 19 03:54 config.txt
drwxrwxr-x 4 sindhusha55 sindhusha55 4096 Mar 19 03:54 ..
drwxrwxr-x 3 sindhusha55 sindhusha55 4096 Mar 19 03:54 .
sindhusha55@instance-1:~/sample_project$ cat .dlv/HEAD.txt
mastersindhusha55@instance-1:~/sample_project$ ls -lart .dlv/master/
total 20
-rw-rw-r-- 1 sindhusha55 sindhusha55    0 Mar 19 03:54 status.txt
drwxrwxr-x 2 sindhusha55 sindhusha55 4096 Mar 19 03:54 stage
drwxrwxr-x 2 sindhusha55 sindhusha55 4096 Mar 19 03:54 commits
-rw-rw-r-- 1 sindhusha55 sindhusha55    0 Mar 19 03:54 commit_log.txt
drwxrwxr-x 2 sindhusha55 sindhusha55 4096 Mar 19 03:54 cache
drwxrwxr-x 3 sindhusha55 sindhusha55 4096 Mar 19 03:54 ..
drwxrwxr-x 5 sindhusha55 sindhusha55 4096 Mar 19 03:54 .
sindhusha55@instance-1:~/sample_project$
```

2) DLV ADD:

It adds the all locally modified files to the “stage” directory in the current branch.

```
sindhusha55@instance-1:~/sample_project$ dlv add
Please enter the directory path
sindhusha55@instance-1:~/sample_project$ dlv add -d Weather-Application/
sindhusha55@instance-1:~/sample_project$ dlv status --json
{
    "Tracked Files": [],
    "Untracked Files": [],
    "Staged Files": [
        "./Weather-Application/angular.json",
        "./Weather-Application/tsconfig.json",
        "./Weather-Application/package.json",
        "./Weather-Application/tslint.json",
        "./Weather-Application/package-lock.json",
        "./Weather-Application/e2e/protractor.conf.js",
        "./Weather-Application/e2e/tsconfig.e2e.json",
        "./Weather-Application/e2e/src/app.po.ts",
        "./Weather-Application/e2e/src/app.e2e-spec.ts",
        "./Weather-Application/src/index.html",
        "./Weather-Application/src/styles.css",
        "./Weather-Application/src/tsconfig.spec.json",
        "./Weather-Application/src/browserslist",
        "./Weather-Application/src/test.ts",
        "./Weather-Application/src/polyfills.ts",
        "./Weather-Application/src/karma.conf.js",
        "./Weather-Application/src/main.ts",
        "./Weather-Application/src/tsconfig.app.json",
        "./Weather-Application/src/favicon.ico",
        "./Weather-Application/src/tslint.json",
        "./Weather-Application/src/environments/environment.ts",
        "./Weather-Application/src/environments/environment.prod.ts",
        "./Weather-Application/src/assets/.gitkeep"
    ],
    "Modified Files": [],
    "Deleted Files": []
}
```

3) DLV COMMIT:

- Creates a snapshot of files and all the file versions are stored in the “cache” directory of the current branch folder.
- It updates few other files with the history like commit_log.txt file
- It saves the current experiment with list of files and their versions.

```
sindhusha55@instance-1:~/sample_project$ dlv commit
Please specify commit message and author name
sindhusha55@instance-1:~/sample_project$ dlv commit -m "m1" -a "a1"
Committed to branch name: master with version: 1
sindhusha55@instance-1:~/sample_project$
```

4) DLV STATUS:

- It compares the files that are changed locally and with the last experiment that is saved.
- It gives the list of modified files, deleted files, tracked files, untracked files.

```
sindhusha55@instance-1:~/sample_project$ dlv status
All files in the current repository and branch: master are up-to-date
sindhusha55@instance-1:~/sample_project$ vim Weather-Application/package-lock.json
sindhusha55@instance-1:~/sample_project$ dlv status --json
{
    "Tracked Files": [
        "./Weather-Application/angular.json",
        "./Weather-Application/tsconfig.json",
        "./Weather-Application/package.json",
        "./Weather-Application/tslint.json",
        "./Weather-Application/e2e/protractor.conf.js",
        "./Weather-Application/e2e/tsconfig.e2e.json",
        "./Weather-Application/e2e/src/app.po.ts",
        "./Weather-Application/e2e/src/app.e2e-spec.ts",
        "./Weather-Application/src/index.html",
        "./Weather-Application/src/styles.css",
        "./Weather-Application/src/tsconfig.spec.json",
        "./Weather-Application/src/browserslist",
        "./Weather-Application/src/test.ts",
        "./Weather-Application/src/polyfills.ts",
        "./Weather-Application/src/karma.conf.js",
        "./Weather-Application/src/main.ts",
        "./Weather-Application/src/tsconfig.app.json",
        "./Weather-Application/src/favicon.ico",
        "./Weather-Application/src/tslint.json",
        "./Weather-Application/src/environments/environment.ts",
        "./Weather-Application/src/environments/environment.prod.ts",
        "./Weather-Application/src/assets/.gitkeep"
    ],
    "Untracked Files": [],
    "Staged Files": [],
    "Modified Files": [
        "./Weather-Application/package-lock.json"
    ],
    "Deleted Files": []
}
sindhusha55@instance-1:~/sample_project$
```

V. PROJECT INCREMENT 2:

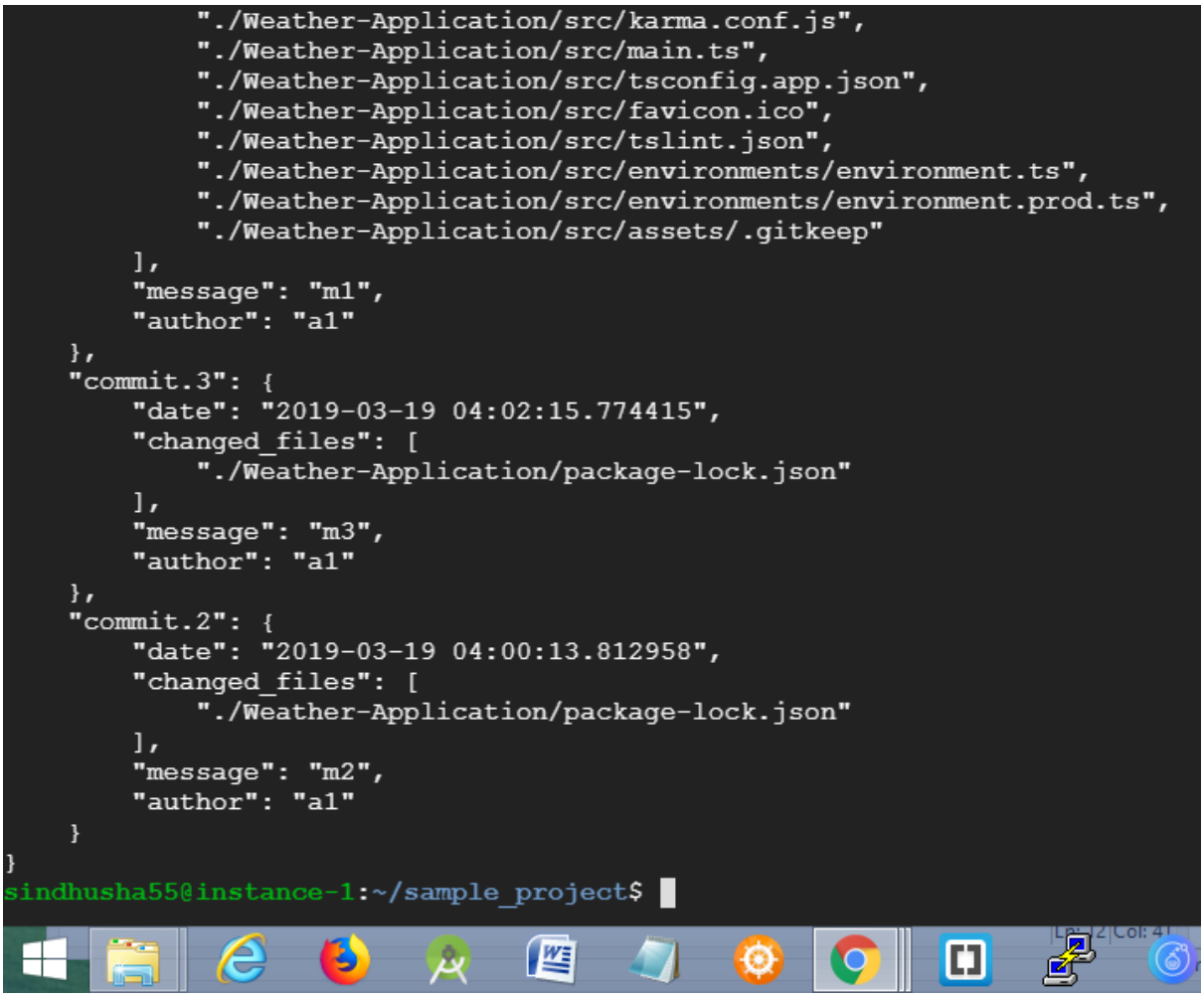
TASKS DONE:

1. Creating Basic User Interface for version control system similar to GIT desktop.
2. Modifying Increment 1 commands to clear all the bugs.
3. Implemented commands {status, history, push, diff of files}
4. Implementing commands {list, desc, diff between models}

DESIGN AND IMPLEMENTATION:

1) DLV HISTORY:

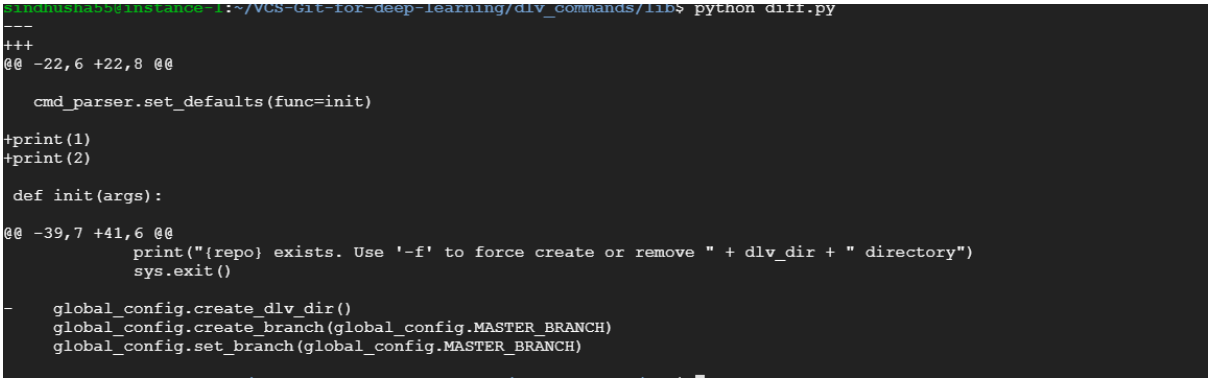
- Displays the content in commit_log.txt file present in the user's current working branch (master).



2) DLV DIFF:

- We are using **"difflib"** to get the diff between two files.

- If user asks for –html option then we are giving the diff output to **diff2html python file** – where it gives the html page with table, line numbers, added lines coloured as green, removed lines coloured as red.
- Displays the diff between two models if user specifies two models.
- Displays the diff between the current model and its previous version if only one model is specified by the user.
- Displays the diff between a file (for current version and the previous version) if only one file is specified by the user.
- Displays the diff between a file (for local repository and the last snapshot version of the file)



3) DLV LIST:

- This command gives list of projects/repositories present in the user account and the details of the repositories like(Project description, author, list of current version files).

4) DLV DESC:

This is similar to DLV LIST command but this command requires extra argument (model name/ project name), where it gives details about only one particular model.

VI. PROJECT INCREMENT 3:

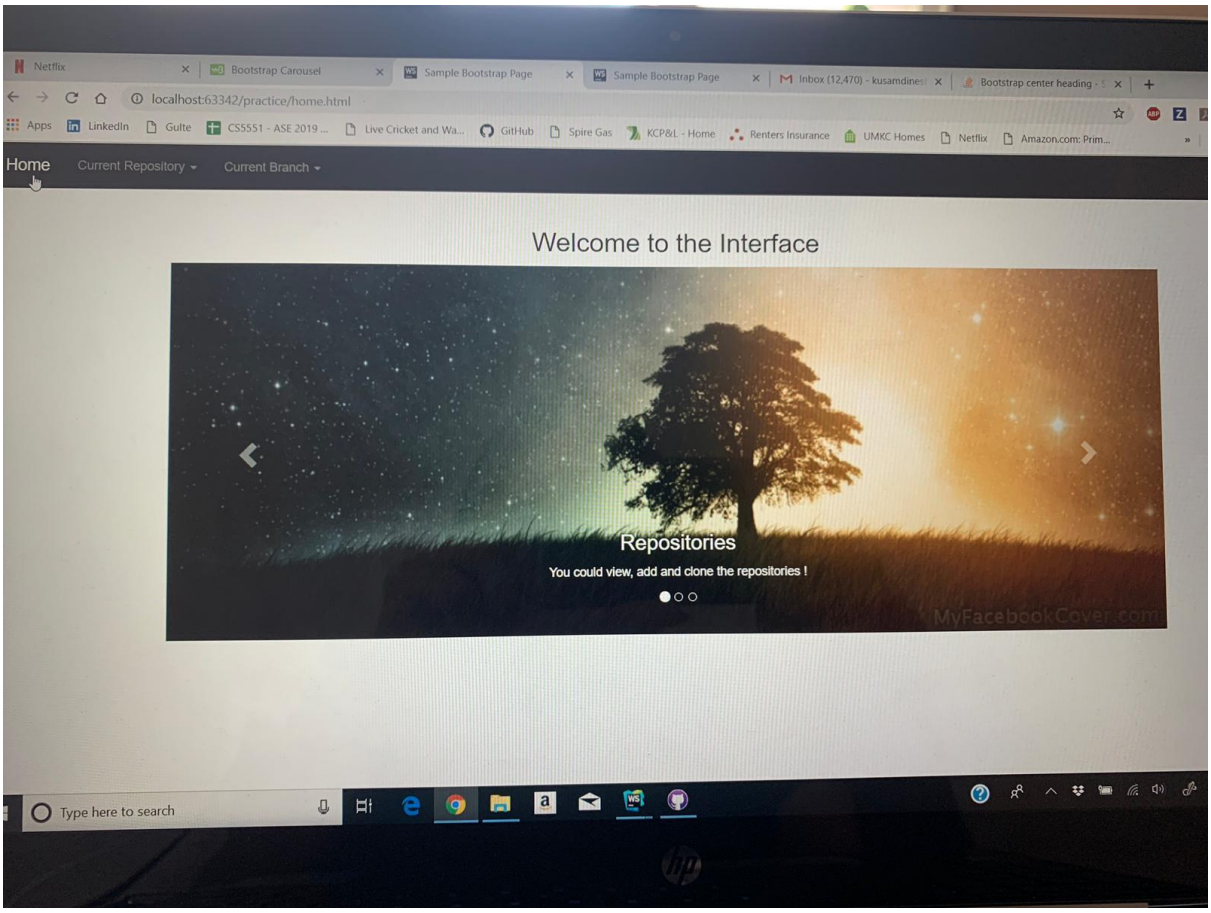
TASKS DONE:

1. Creating Basic User Interface for Deep learning Version control system.
2. User can Register and Authenticate into DLV.
3. After Login User can upload files/repositories to the server.
4. After login User can view List of repositories.
5. On click of the repository -> It shows the status and history of the files in the repository.

DESIGN AND IMPLEMENTATION:

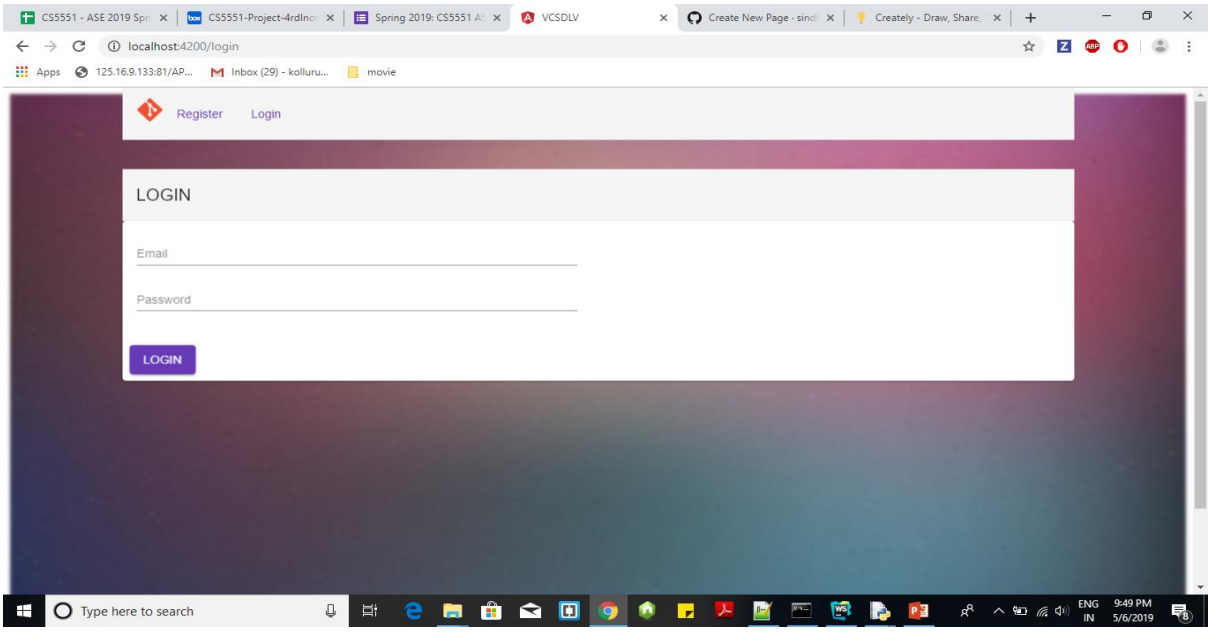
Attaching the screenshots of the User Interface:

1) Created a static page with the demo of the website

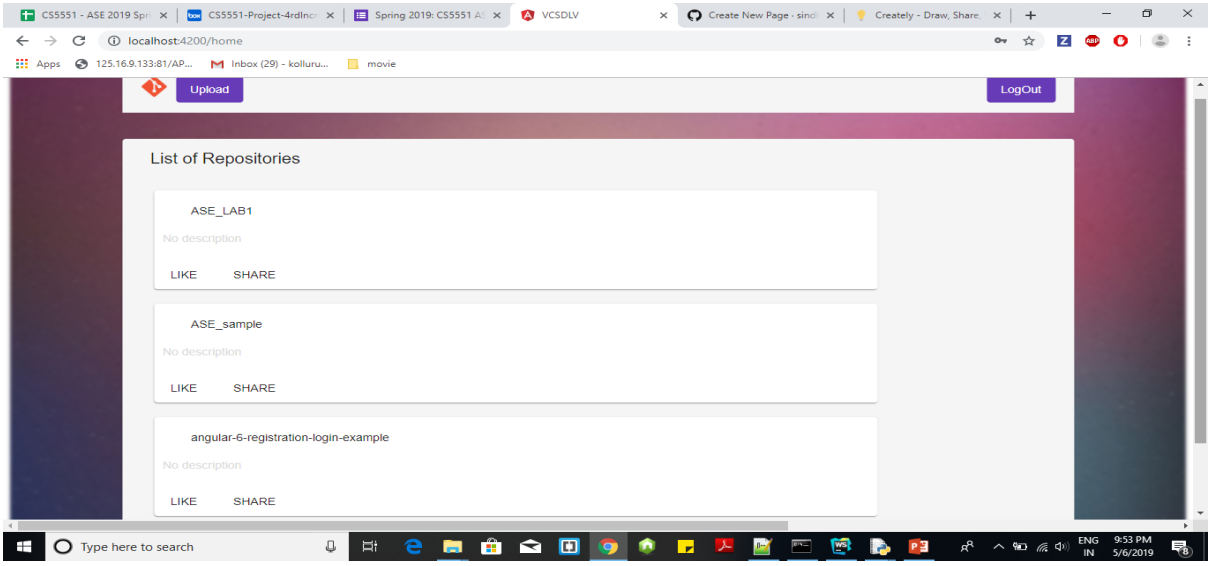


2) Created registration page where user can register the details and the details are stored in the MongoDB.

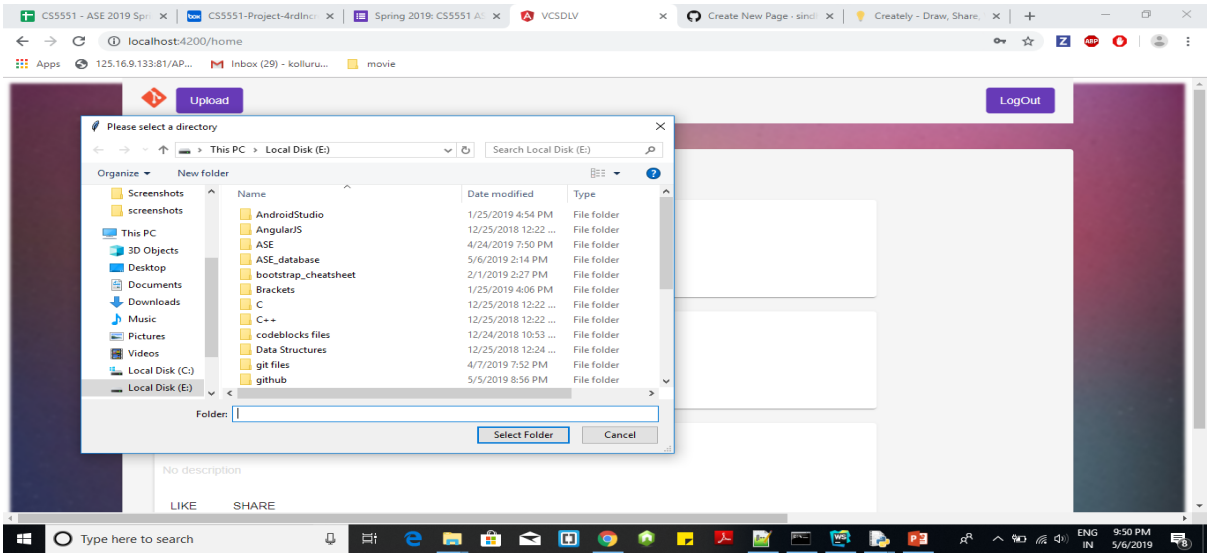
3) Created login page where user can log in to the page and the user credentials are verified by connecting to the MongoDB.



4) After logging in, User can view the list of repositories present in the user account.
    - In this **“dlv list”** command is executed and the result is displayed in the UI.



5) User can upload the local project to the server. Here the server is “**Local File System**”.
    - Where when user uploads the project folder – folder is saved in the separate server folder located in the local file system instead of storing in the online cloud.
    - **“Dlv init”** and **“dlv commit”** commands are executed. (this creates the initial version control system)



6) User can logout of the Application.

VII. PROJECT INCREMENT 4:

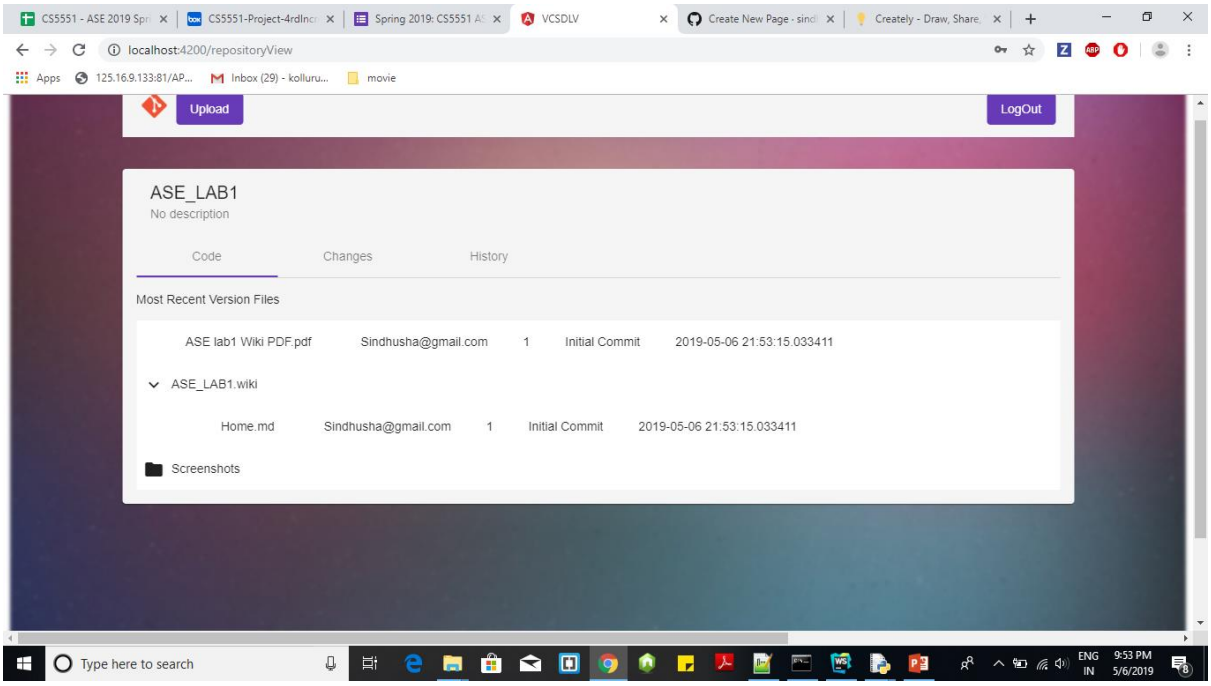
TASKS DONE:

1. Improving the User Interface and to clear all console errors
2. After login User can view List of repositories.
3. On click of the repository -> It shows the status and history of the files in the repository.
4. Frequent polling of status GET request for every 30sec to get list of modified files.
5. Updating all details about the repository (list, status, history) after user clicks the commit.

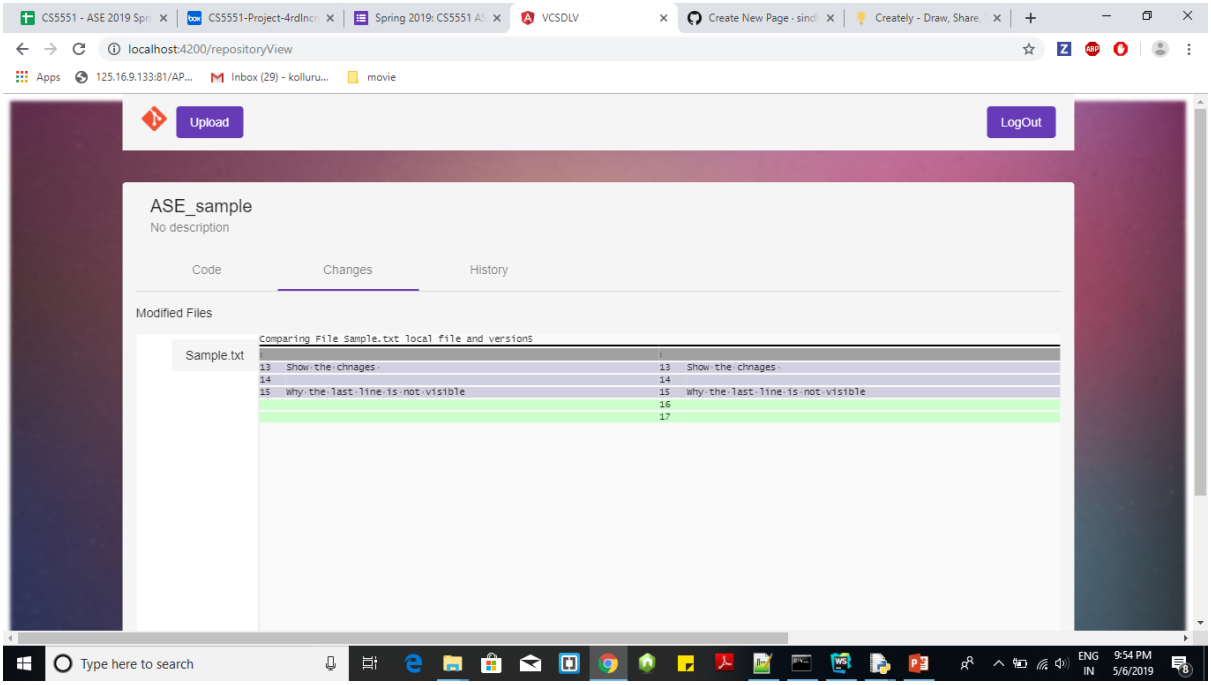
DESIGN AND IMPLEMENTATION:

When user clicks on particular repository –

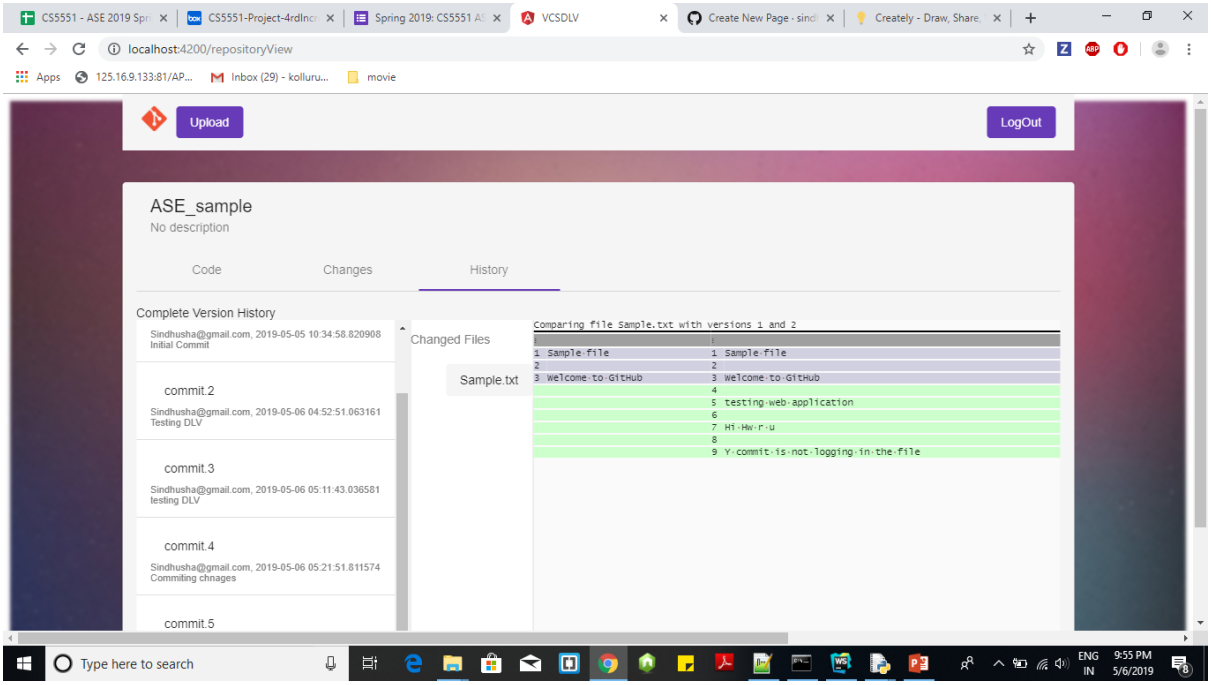
1) **Code TAB:** it shows the list of all files from the last experiment.
   - It shows details like file-name, folder-name, author, commit message, date
   - It executes "**dlv list"** and the result is displayed in the UI.



2) **History TAB:** It shows the list of all experiments in the project.
   - User can view different experiments.
   - For each experiment – user can view list of changed files.
   - User can click on any file and can view the difference of files that are changed in the content.
   - It executes **"dlv history"** command and displays the result in UI.



3) **Changes TAB:** It shows the list of changes done to the local repository and the last experiment.
    - It executes the **"dlv status"** command for every 30seconds and updates the results in the UI



VIII. FUTURE WORK:

1) Creating executable file from the python dlv commands for command line interface.

2) Implementing search command (across users –complete database)

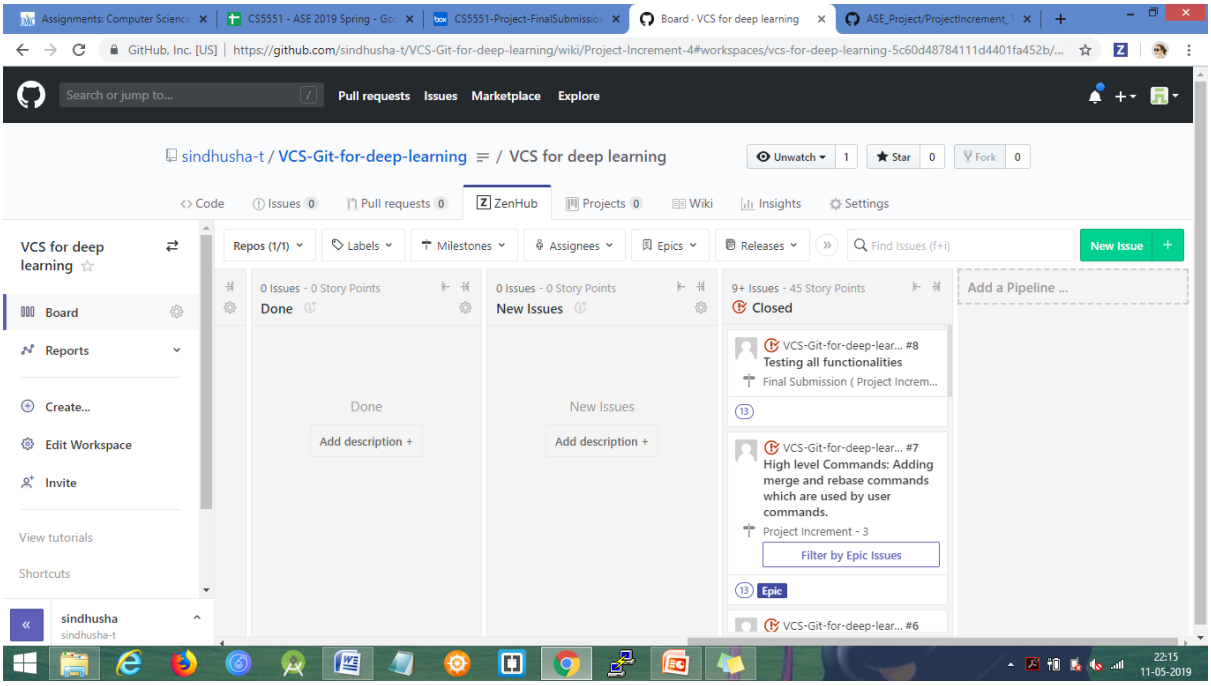
3) Reverting back to the previous changes (complete folder just like tags and branches in GIT).

IX. PROJECT DEPLOYMENT:

1) We deployed the backend server for validating the user to heroku.
URL: https://aseprojectapp.herokuapp.com/user

2) As we are using local file system and tkinter library we were unable to deploy the dlv commands written in python to heroku.

3) As Web application has dependency with python commands – we are not deploying the angular code to heroku.



X. PROJECT MANAGEMENT:

Equal Individual Contribution – 25% each.

- Every member of the project contributed solely for the project.
- We have discussed at every stage about the design, implementation, and how to divide the work.
- We divided the work and followed the Agile process to complete the project.
- We spend some time together to clear each other doubts and integrate each one modules into a single project.

The work has been completed under the guidance of

Dr. Yugi Lee,
Rajaram Anantharaman,
and TAs (Sirisha Rella, Bhargavi Nadendla)

in CS5551 Advanced Software Engineering,
University of Missouri - Kansas City), spring 2019.