

## **Experiment 8: Change specification and use any SCM Tool to make different versions for the project**

**Learning Objective:** Students will able to create versions using Github tool

**Tools:** Github

### **Theory:**

Software configuration management: The traditional software configuration management (SCM) process is looked upon by practitioners as the best solution to handling changes in software projects. It identifies the functional and physical attributes of software at various points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development life cycle.

Software configuration management is a part of software engineering, which focuses mainly on maintaining, tracking and controlling the changes done to the software configuration items.

Configuration management is present in all phase of software development. The configuration items can be all the objects which come as an output of the development process e.g. coding phase produces source code, exes and obj files. The various configuration items can be:

1. Source code,
2. Documents
3. Data used in the programs

In Configuration management, there can be multiple versions created for any configuration item (Source code/ documents). Each version can be identified by unique configuration or an attribute which is associated with each version. E.g. the version number.

### **Terminologies used in version control**

1. SCI – Software configuration items, i.e. the documents and code which will be having version number and saved.
2. Repository- it is the system where all the SCIs will be stored.
3. Check in- to store the tested and qualified source code.
4. Checkout- to get a copy of the stored SCI from the repository.

5. Add – Add to the local repo and keep ready for commit
6. Commit- to save the file in repository and create a version

#### Advantages

- 1) The versions are stored in the repository; hence they are available as backups.
- 2) Multiple people can work simultaneously on same files/source code, without losing the changes made by other developers
- 3) It is easy to find the files with specifications as a versions are stored with version numbers

GitHub offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a Web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as bug tracking, feature requests, task management for every project.

**Learning Outcomes:** Students should have the ability to

**LO1:** to understand the need of doing configuration management.

**LO2:** Identify the dissimilarity between version and variant

**LO3:** provide the knowledge of the benefits of using version control

**LO4:** To understand the types of version control system

### Initializing a Repository

```
fa327@Faraz MINGW64 ~/Desktop/SE
$ git init
Initialized empty Git repository in C:/Users/fa327/Desktop/SE/.git/

fa327@Faraz MINGW64 ~/Desktop/SE (master)
$ git status
On branch master
```

### Creating HTML Branch

```
fa327@Faraz MINGW64 ~/Desktop/SE (master)
$ git checkout -b html
Switched to a new branch 'html'
```

### Configuring Username and Email

```
fa327@Faraz MINGW64 ~/Desktop/SE (html)
$ git config --global user.email 'fa32729@gmail.com'

fa327@Faraz MINGW64 ~/Desktop/SE (html)
$ git config --global user.name 'Faraz'
```

### Committing the HTML code

```
fa327@Faraz MINGW64 ~/Desktop/SE (html)
$ git add .

fa327@Faraz MINGW64 ~/Desktop/SE (html)
$ git commit -m 'Added the HTML'
[html (root-commit) 0ec2816] Added the HTML
1 file changed, 12 insertions(+)
create mode 100644 html/index.html
```

### Creating a CSS repository and Committing it

```
fa327@Faraz MINGW64 ~/Desktop/SE (html)
$ git checkout -b 'CSS'
Switched to a new branch 'CSS'

fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$ git status
On branch CSS
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    css/

nothing added to commit but untracked files present (use "git add" to track)

fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$ git add .

fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$ git commit -m 'Added the CSS'
[CSS 69fbaaa] Added the CSS
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 css/styles.css
```

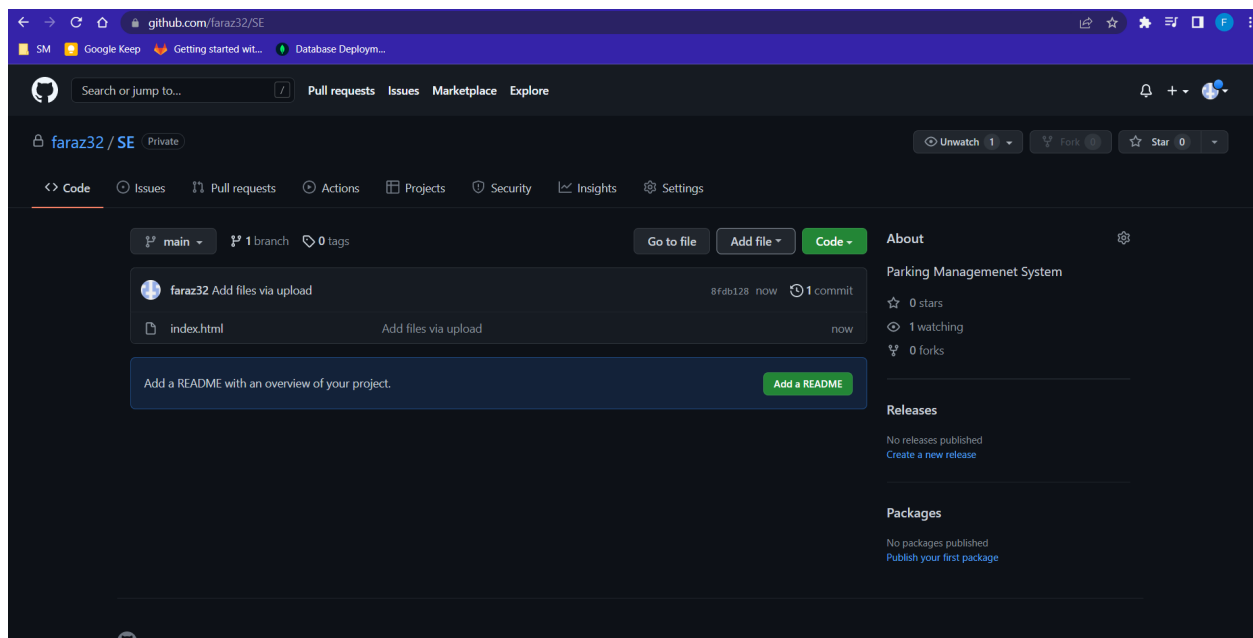
## All the branches of the project

```
fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$ git status
On branch CSS
nothing to commit, working tree clean

fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$ git branch -a
* CSS
  html

fa327@Faraz MINGW64 ~/Desktop/SE (CSS)
$
```

## Repository



The screenshot shows a GitHub repository page for 'faraz32/SE'. The repository is private and has 1 branch (main) and 0 tags. The main branch has 1 commit (8fdb128) and 1 file (index.html). The repository is titled 'Parking Management System' and has 0 stars, 1 watching, and 0 forks. The page includes sections for 'About', 'Releases', and 'Packages'. The 'About' section contains the title 'Parking Management System' and the description 'No releases published'. The 'Releases' section contains the text 'No releases published' and a link to 'Create a new release'. The 'Packages' section contains the text 'No packages published' and a link to 'Publish your first package'.

**Outcomes:** Upon completion of the course students will be able to create versions for the project.

**Conclusion:** Version were created using Github

**Viva Questions:**

1. What is difference between git and Github?
2. What is version control? Why is it required?
3. What are other tools for version control?
4. What are different types of version control?

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [ 40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				