ADVANCED WEB TECHNOLOGIES LABORATORY

Lab Manual

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted by:

Shivay Bhandari (D3 IT-B1)

CRN:1921142

URN: 1905398

Submitted to:

Er. Manjot Kaur

Assistant Professor

Department of Information Technology,

Guru Nanak Dev Engineering College,

Ludhiana-141006

INDEX

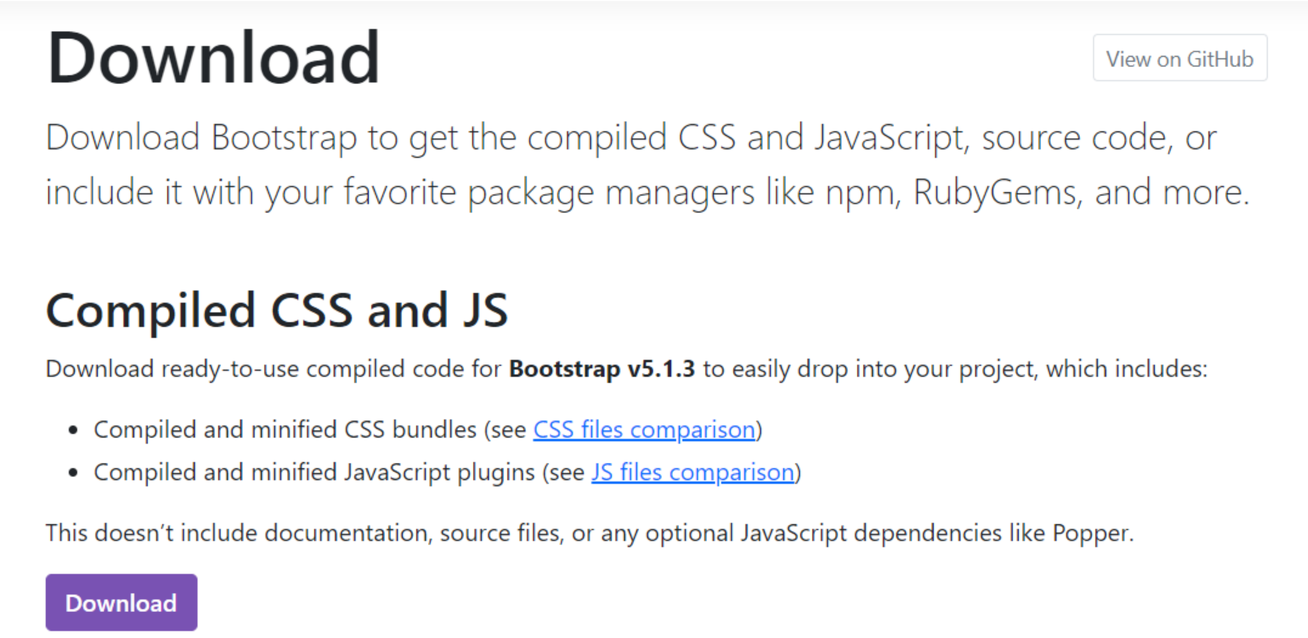
Practical - 1

TO INSTALL AND SETUP THE HTML5 BASED BOOTSTRAP FRAMEWORK TO DEPLOY BASIC HTML ELEMENTS USING BOOTSTRAP CSS

Step 1. Go to Google and type Bootstrap. The Bootstrap links will open

Step 2. Click the first link in the search results

Step 3. The bootstrap homepage will be opened. Click on “Download Bootstrap” button



Download

View on GitHub

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.

Compiled CSS and JS

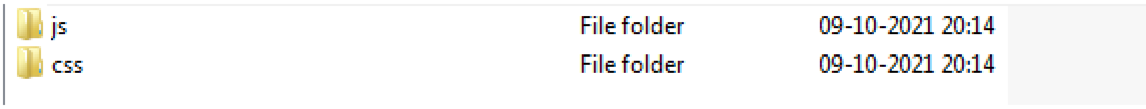
Download ready-to-use compiled code for **Bootstrap v5.1.3** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see CSS files comparison)
- Compiled and minified JavaScript plugins (see JS files comparison)

This doesn’t include documentation, source files, or any optional JavaScript dependencies like Popper.

Download

Step 4. The page will be opened to select the Bootstrap option. The file be downloaded in zip format. You should extract the files. Inside the file there are three kinds of file available



js	File folder	09-10-2021 20:14
css	File folder	09-10-2021 20:14

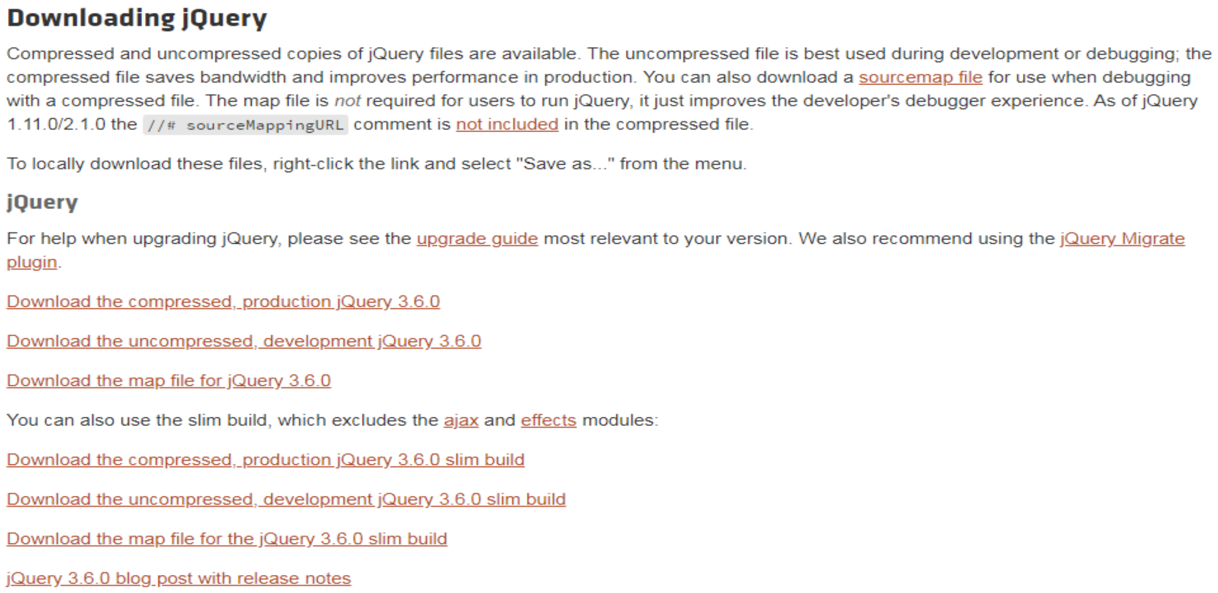
Step 5. Go to Google and type jquery.com

Step 6. Click the first option as shown in the image. The jQuery page will be opened. Click the “Download jQuery” option

Step 7. The jQuery file will be opened

Step 8. There are a number of download links available. Select advanced version to download

Step 9. The file will be download. In that file, copy and paste the JS folder. That is the way to include jquery.js files in the Bootstrap folder



Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. You can also download a sourcemap file for use when debugging with a compressed file. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the `//# sourceMappingURL` comment is not included in the compressed file.

To locally download these files, right-click the link and select "Save as..." from the menu.

jQuery

For help when upgrading jQuery, please see the upgrade guide most relevant to your version. We also recommend using the jQuery Migrate plugin.

Download the compressed, production jQuery 3.6.0

Download the uncompressed, development jQuery 3.6.0

Download the map file for jQuery 3.6.0

You can also use the slim build, which excludes the ajax and effects modules:

Download the compressed, production jQuery 3.6.0 slim build

Download the uncompressed, development jQuery 3.6.0 slim build

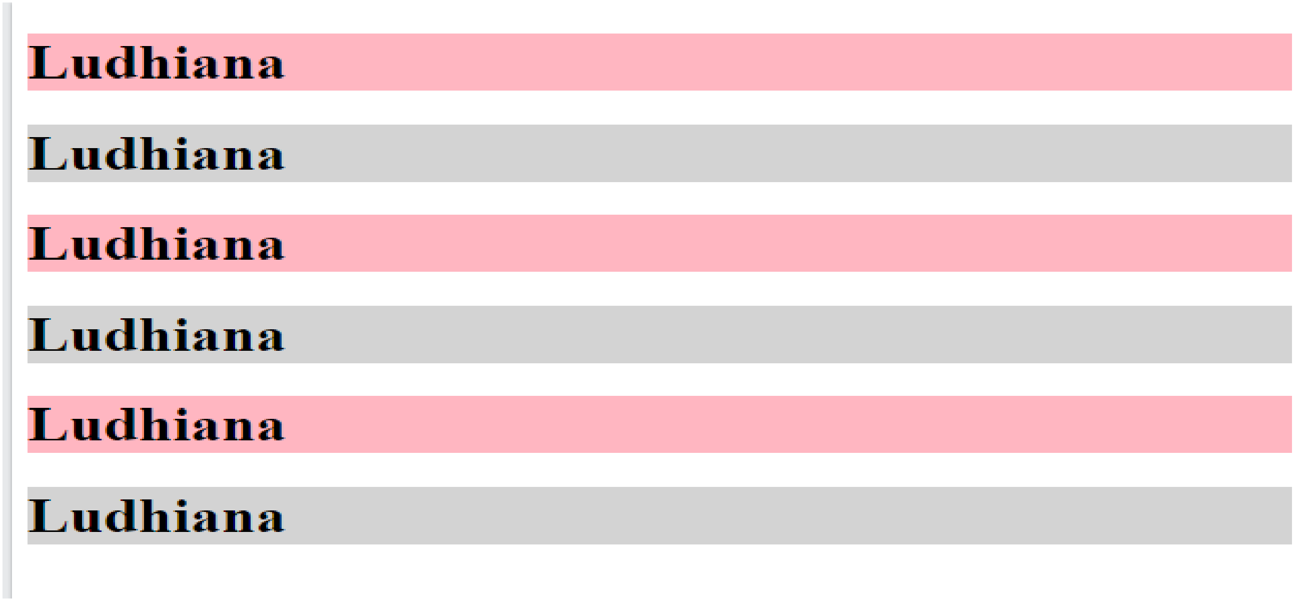
Download the map file for the jQuery 3.6.0 slim build

jQuery 3.6.0 blog post with release notes

Step 10. After saving the file wherever you want, open the Notepad++. Write HTML codes for linking all the files in your HTML page

Step 11. After writing the codes, it must be saved where the Bootstrap file for ex-index.html

Step 12. After saving the files, you should double click the file and the HTML page will run.



Ludhiana

Ludhiana

Ludhiana

Ludhiana

Ludhiana

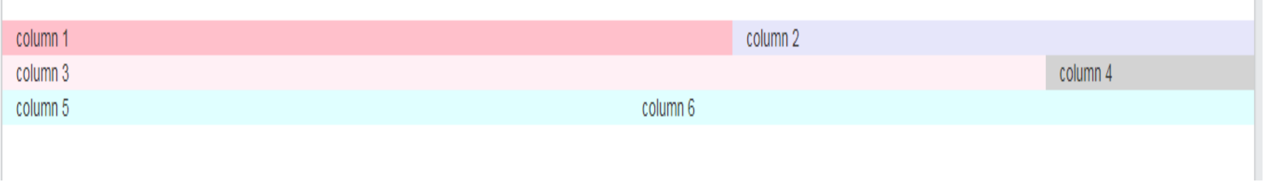
Ludhiana

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Bootstrap</title>

    <link rel="stylesheet" href="./bootstrap-5.1.3-dist/css/bootstrap.css" />

  </head>

  <body>

  <h1 style="background-color:lightpink;">

    <div class="container fs-1">
```

```
      <div class="row akign-items-start my-5">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div></h1>
          <h1 style="background-color:lightgrey;">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div></h1>
      </div>
        <h1 style="background-color:lightpink;">
      <div class="row akign-items-start my-5">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div></h1>
          <h1 style="background-color:lightgrey;">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div>
      </div></h1>
        <h1 style="background-color:lightpink;">
      <div class="row akign-items-start my-5">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div></h1>
          <h1 style="background-color:lightgrey;">
        <div class="col bg-primary text-white text-capitalize">Ludhiana</div>
        </h1>
      </div>
    </div>
    <script src="./bootstrap-5.1.3-dist/js/bootstrap.js"></script>
  </body>
</html>
```

Practical 2

To understand and deploy multicolumn grid layout in Bootstrap



column 1 column 2

column 3 column 4

column 5 column 6

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>Hello World!</h1>
  <p>Resize the browser window to see the effect.</p>
  <div class="row">
    <div class="col-xs-9 col-md-7" style="background-color:pink;">column 1</div>
    <div class="col-xs-3 col-md-5" style="background-color:lavender;">column 2</div>
  </div>
  <div class="row">
    <div class="col-xs-6 col-md-10" style="background-color:lavenderblush;">column 3</div>
    <div class="col-xs-6 col-md-2" style="background-color:lightgrey;">column 4</div>
  </div>
  <div class="row" style="background-color:lightcyan;">
    <div class="col-xs-6">column 5</div>
    <div class="col-xs-6">column 6</div>
  </div>
</div>


</body>
</html>
```

Understanding Bootstrap Grid System

- Bootstrap grid system classifies screen sizes into four types: Extra-small, Small, Medium and Large.
- The Extra-small is for mobile phones. The Small is for tablets, like I-Pad. The Medium is for desktop monitors. And the Large is just for larger desktop monitors.

- To help you fine tune responsiveness for these four types of screens, Bootstrap divides the width of web viewing devices into the 12-column grid system.
- Every one of those four Bootstrap Grid types becomes responsive only if you assign to it its own designated CSS class.

PRACTICAL 3

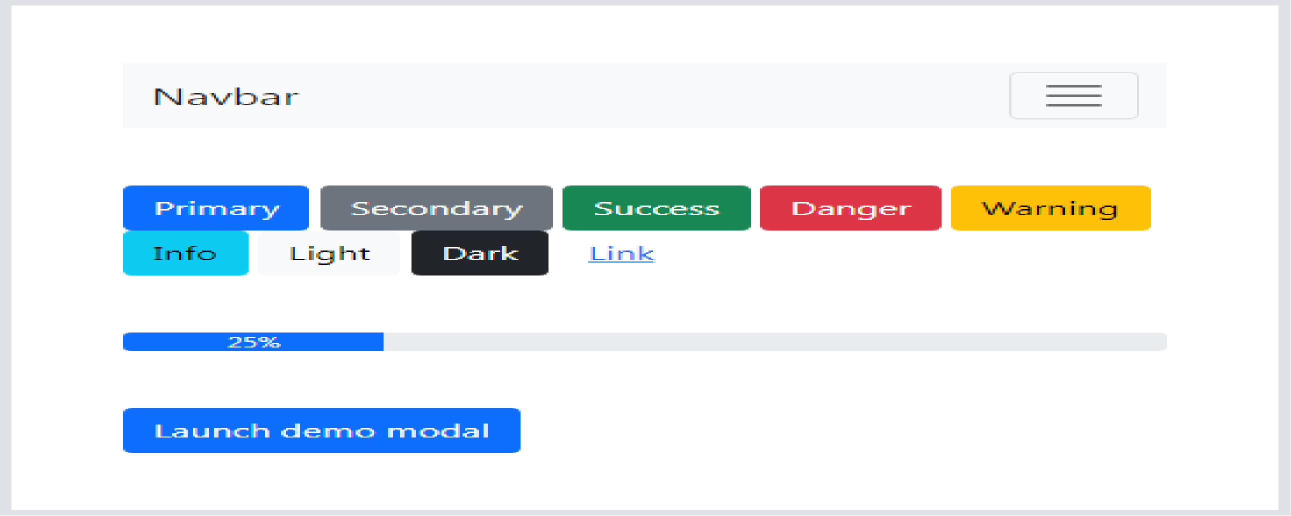
TO DEPLOY DIFFERENT TYPES OF BUTTON, PROGRESS BARS, MODALS AND NAVIGATION BARS USING BOOTSTRAP

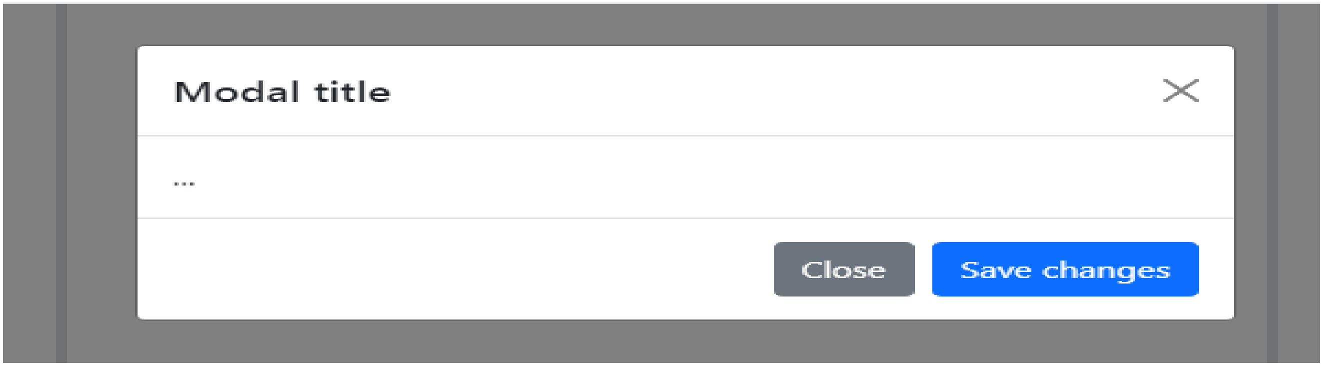
```
 <!DOCTYPE html>
<html lang="en">
   <head>
     <meta charset="UTF-8" />
     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
     <title>Bootstrap</title>
     <link rel="stylesheet" href="./bootstrap-5.1.3-dist/css/bootstrap.css" />
   </head>
   <body>
     <div class="border border-5 mx-5 my-5">
       <!-- Navbar -->
       <nav class="navbar navbar-expand-lg navbar-light bg-light mx-5 my-5">
         <div class="container-fluid">
           <a class="navbar-brand" href="#">Navbar</a>
           <button   >
            class="navbar-toggler"
             type="button"
             data-bs-toggle="collapse"
             data-bs-target="#navbarSupportedContent"
             aria-controls="navbarSupportedContent"
             aria-expanded="false"
             aria-label="Toggle navigation"
             <span class="navbar-toggler-icon"></span>
           </button>
           <div class="collapse navbar-collapse" id="navbarSupportedContent">
             <ul class="navbar-nav me-auto mb-2 mb-lg-0">
```

```
  <li class="nav-item">
    <a class="nav-link active" aria-current="page" href="#">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item dropdown">
    <a
      class="nav-link dropdown-toggle"
      href="#"
      id="navbarDropdown"
      role="button"
      data-bs-toggle="dropdown"
      aria-expanded="false"
    >
      Dropdown
    </a>
    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
      <li><a class="dropdown-item" href="#">Action</a></li>
      <li><a class="dropdown-item" href="#">Another action</a></li>
      <li><hr class="dropdown-divider" /></li>
      <li>
        <a class="dropdown-item" href="#">Something else here</a>
      </li>
    </ul>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled">Disabled</a>
  </li>
</ul>
<form class="d-flex">
```

```
        <input
          class="form-control me-2"
          type="search"
          placeholder="Search"
          aria-label="Search"
        />
        <button class="btnbtn-outline-success" type="submit">
          Search
        </button>
      </form>
    </div>
  </div>
</nav>

<!-- Buttons -->
<div class="my-5 mx-5">
  <button type="button" class="btnbtn-primary">Primary</button>
  <button type="button" class="btnbtn-secondary">Secondary</button>
  <button type="button" class="btnbtn-success">Success</button>
  <button type="button" class="btnbtn-danger">Danger</button>
  <button type="button" class="btnbtn-warning">Warning</button>
  <button type="button" class="btnbtn-info">Info</button>
  <button type="button" class="btnbtn-light">Light</button>
  <button type="button" class="btnbtn-dark">Dark</button>
  <button type="button" class="btnbtn-link">Link</button>
</div>

<!-- Progress Bar -->
<div class="progress my-5 mx-5">
  <div
    class="progress-bar"
    role="progressbar"
    style="width: 25%"
    aria-valuenow="25"
    aria-valuemin="0"
    aria-valuemax="100"
  >
    25%
  </div>
</div>

<!-- Modal -->
<div class="my-5 mx-5">
  <button
    type="button"
    class="btnbtn-primary"
    data-bs-toggle="modal"
    data-bs-target="#exampleModal"
  >
    Launch demo modal
  </button>

  <div
    class="modal fade"
    id="exampleModal"
    tabindex="-1"
    aria-labelledby="exampleModalLabel"
    aria-hidden="true"
  >
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
```

```
                <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
                <button
                  type="button"
                  class="btn-close"
                  data-bs-dismiss="modal"
                  aria-label="Close"
                ></button>
              </div>
              <div class="modal-body">...</div>
              <div class="modal-footer">
                <button
                  type="button"
                  class="btnbtn-secondary"
                  data-bs-dismiss="modal"
                >
                  Close
                </button>
                <button type="button" class="btnbtn-primary">
                  Save changes
                </button>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <script src="./bootstrap-5.1.3-dist/js/bootstrap.js"></script>
  </body>
</html>
```





1.) Button:

The .btn classes are designed to be used with the <button> element. However, you can also use these classes on <a> or <input> elements (though some browsers may apply a slightly different rendering).

When using button classes on <a> elements that are used to trigger in-page functionality (like collapsing content), rather than linking to new pages or sections within the current page, these links should be given a role="button" to appropriately convey their purpose to assistive technologies such as screen readers.

2.) Navigation:

- Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl} for responsive collapsing and color scheme classes.

- Navbars and their contents are fluid by default. Use optional containers to limit their horizontal width.
- Use our spacing and flex utility classes for controlling spacing and alignment within navbars.
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behaviour depends on our Collapse JavaScript plugin.
- Navbars are hidden by default when printing. Force them to be printed by adding .d-print to the .navbar. See the display utility class.
- Ensure accessibility by using a <nav> element or, if using a more generic element such as a <div>, add a role="navigation" to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

3.) Progress Bar:

Progress components are built with two HTML elements, some CSS to set the width, and a few attributes. We don’t use the HTML5 <progress> element, ensuring you can stack progress bars, animate them, and place text labels over them.

- We use the .progress as a wrapper to indicate the max value of the progress bar.
- We use the inner .progress-bar to indicate the progress so far.
- The .progress-bar requires an inline style, utility class, or custom CSS to set their width.
- The .progress-bar also requires some role and aria attributes to make it accessible.

4.) Modal:

- Modals are built with HTML, CSS, and JavaScript. They are positioned over everything else in the document and remove scroll from the <body> so that modal content scrolls instead.
- Clicking on the modal “backdrop” will automatically close the modal.

- Bootstrap only supports one modal window at a time. Nested modals are not supported as we believe them to be poor user experiences.
- Modals use position: fixed, which can sometimes be a bit particular about its rendering. Whenever possible, place your modal HTML in a top-level position to avoid potential interference from other elements. You’ll likely run into issues when nesting a .modal within another fixed element.
- Due to how HTML5 defines its semantics, the autofocus HTML attribute has no effect in Bootstrap modals. To achieve the same effect, use some custom JavaScript:

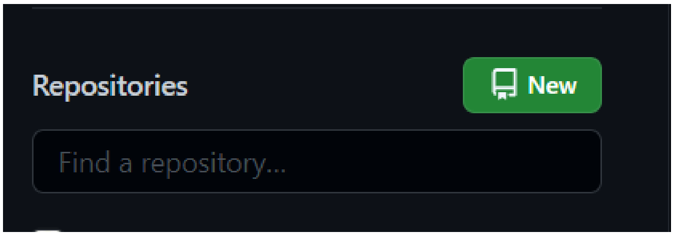
PRACTICAL - 4

TO CREATE AND SETUP THE GIT REPOSITORY ON BITBUCKET OR GITHUB USING SSH

Create the Git repository on GitHub using SSH

1.) Create a GitHub account, if you have one login into your account

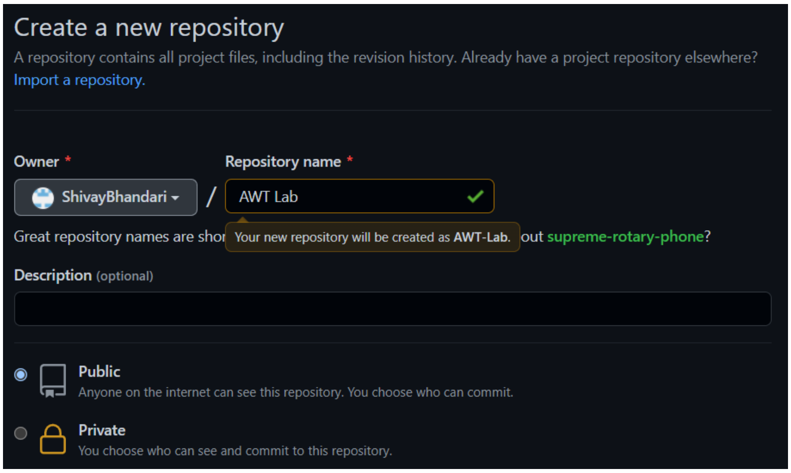
2.) Click the new repository button to create a repository



3.) Enter a Name for the repository, enter the description (optional)

4.) Choose private or public repository according to your specifications

5.) Click on the create repository button



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * ShivayBhandari / Repository name * AWT Lab ✓

Great repository names are shor Your new repository will be created as AWT-Lab. out supreme-rotary-phone?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

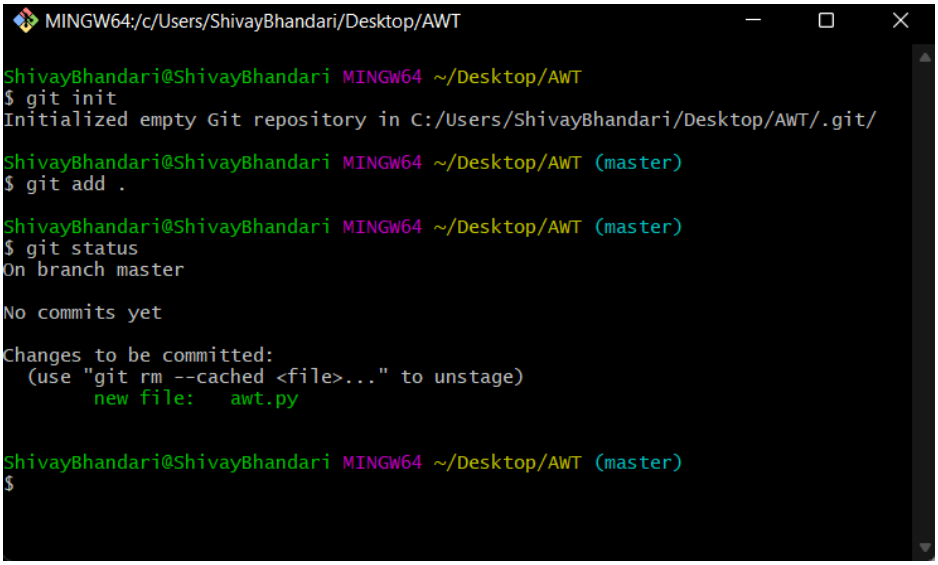
6.) Use SSH keys for Git authorization, this means you have to supply a username and password to be able to write to this repository

PRACTICAL - 5

TO PERFORM PUSH, CLONE AND PATCH OPERATIONS IN GIT REPOSITORY

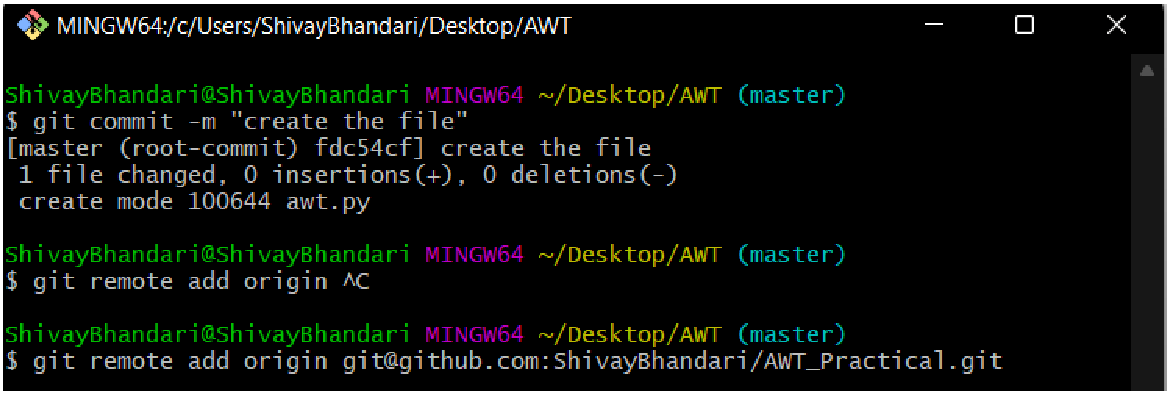
- **GIT PUSH➔**The git push command is used to transfer or push the commit, which is made on a local branch in your computer to a remote repository like GitHub. The command used for pushing to GitHub is given below.
  - git push 'remote_name' 'branch_name

1. Creating a new repository.
2. Open your Git Bash
3. Create your local project in your desktop directed towards a current working directory and initialize the git repository
   - Move to the specific path in your local computer by cd 'path_name'
   - Use git init to initialize the repository. It is used to create a new empty repository or directory consisting of files with the hidden directory.
4. Add the file to the new local repository.
   - Use git add . in your bash to add all the files to the given folder.
   - Use git status in your bash to view all the files which are going to be staged to the first commit.



```
MINGW64:/c/Users/ShivayBhandari/Desktop/AWT

ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT
$ git init
Initialized empty Git repository in C:/Users/ShivayBhandari/Desktop/AWT/.git/

ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git add .

ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   awt.py


ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$
```

5. Commit the files staged in your local repository by writing a commit message.
   - You can create a commit message by git commit -m 'your message', which adds the change to the local repository.

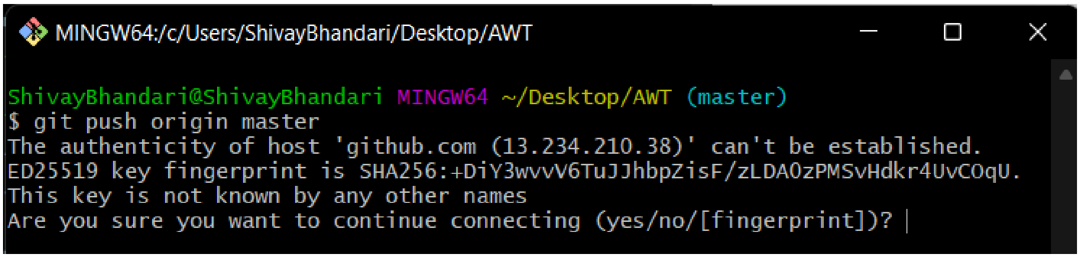


```
ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git commit -m "create the file"
[master (root-commit) fdc54cf] create the file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 awt.py

ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git remote add origin ^C

ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git remote add origin git@github.com:ShivayBhandari/AWT_Practical.git
```

6. Copy your remote repository's URL from GitHub.
   - The HTTPS or SSH or URL is copied from the given GitHub account, which is the place of the remote repository.
   - Add the URL copied, which is your remote repository to where your local content from your repository is pushed.

7 .Push the code in your local repository to GitHub

- git push -u origin main is used for pushing local content to GitHub.
- the origin is your default remote repository name and '-u' flag is upstream, which is equivalent to '-set-upstream.' and the main is the branch, name. Upstream is the repository that we have cloned the project.

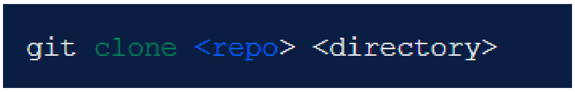


```
ShivayBhandari@ShivayBhandari MINGW64 ~/Desktop/AWT (master)
$ git push origin master
The authenticity of host 'github.com (13.234.210.38)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

7. View your files in your repository hosted on GitHub.

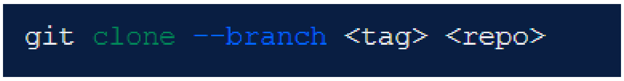
- **GIT CLONE➔**git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols.

1. Cloning to a specific folder. Clone the repository located at <repo> into the folder called ~<directory>! on the local machine.



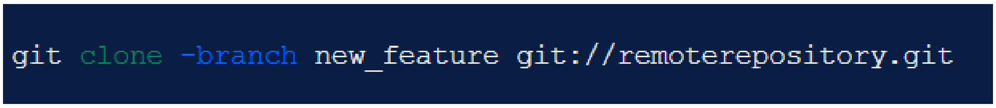
```
git clone <repo> <directory>
```

2. Cloning a specific tag. Clone the repository located at <repo> and only clone the ref for <tag>.



```
git clone --branch <tag> <repo>
```

3. git clone -branch. The -branch argument lets you specify a specific branch to clone instead of the branch the remote HEAD is pointing to, usually the main branch. In addition, you can pass a tag instead of branch for the same effect.



```
git clone -branch new_feature git://remoterepository.git
```

- **GIT PATCH➔** To create a Git patch file, you have to use the “git format-push” command, specify the branch and target directory where you want your patches to be stored

  Creating a patch

  - There are two easy ways to create a patch file. You can use diff command and compare a original with a modified file or you can use the Git to output a patch based on one or more commit

  Diff command

  - To create a patch file using the diff and previous example, duplicate the file you are changing with a new name, and make the change to old one

Practical 6

To install and setup the Codelgniter Framework and to understand its MVS architecture

Codelgniter is installed in four steps:

I Unzip the package.
2. Upload the Codelgniter folders and files to your server. Normally the index.php file will be at your root.
3 Open the application/config/config.php file with a text editor and set your base URI.
If you intend to use encryption or sessions, set your encryption key.

4. If you intend to use a database. open the application/config/database.php file with a text editor and set your database settings.

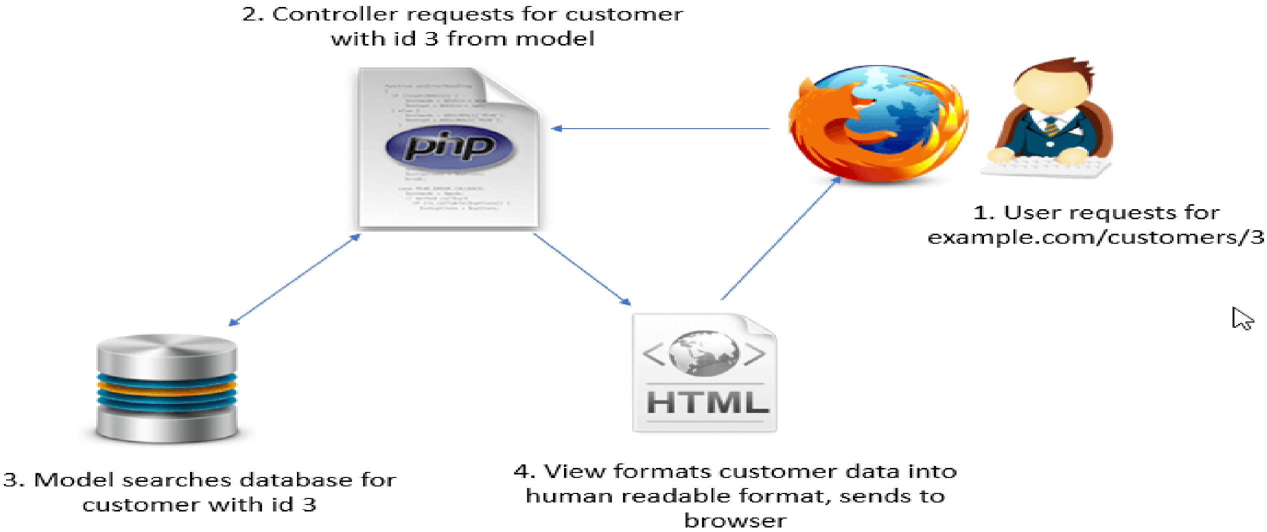
M VC Standards for Model-View -Control.

It is an architectural pattern that splits the application into three major components.

l. ,Model deals with business logic and database interactions

2. Controller coordinates the activities between the model and the view

3. View is responsible for data presentation


2. Controller requests for customer with id 3 from model
php
1. User requests for example.com/customers/3
HTML
3. Model searches database for customer with id 3
4. View formats customer data into human readable format, sends to browser

Practical 7

To construct a simple login page web application to authenticate users using Codelgniter Framework and also Perform CURD operations

I Create Database Table

To store the user account information. a table is required in the database using MySql

2. Config (autoload.php)

In this Codelgniter Login System script, the built-in system library and helper are used. Define the frequently used libraries and helpers in the application/config /autoload.php file.

3. Controller (Users.php)

The Users controller handles the user registration and authentication-related operations.

- ➢ construct() —Load the Form validation library and User model.
- ➢ Get the user's login status from SESSION.
- ➢ index() — Based on the login status redirects the user to the account/login pager
- ➢ account() —
  - Get the logged-in user's account information using getRows() method of User model.
  - Pass user data and load the account details view.
- ➢ login() –
  - Initially, load the login form view.
  - If the form is submitted,

    a. Validate the posted fields data using Codelgniter Form Validation library.

    b .Check whether the user exists in the database with the provided login credentials.

    c. Based on the login status redirects the user to account/ login page.
- ➢ registration() —

Initially, load the registration form view

- ➢ If the form is submitted,
- ➢ Validate the posted fields data using Codelgniter Form Validation library.
- ➢ Check whether the provided email already exists in the database using a custom call back function (email check).

(vii).

- Insert the user's account information in the database using insert() method of the User model.
- logout( )-log the user out from their account.

- email_check() - Custom callback function used with CodeIgniter from validation library to check the existing email.

Model (User.php)

The User model handles the database related operations (fetch and insert).

- getRows() - Fetch the user data the database based on the conditions.
- insert() - Insert the user account data in the database.

View

- elements/ This directory holds the element parts of the web pages.
- elements/ header.php This file holds the header part of the web pages.

users/

(i). This directory holds the view files of the user login system.

users/registration.php

(ii). This view file is used to display the registration form.

(iii). This file contains form HTML to collect the user account information. Once the user submits the form, it is submitted to the registration() method of Users controller.

userslogin.php

(i). This view file is used to display the login form.

(ii). This file contains login form HTML to authenticate the user their account. once the user submits the form , it is submitted to the login() method of Users controller.

users/account.php

(i). This view displays the account details of the logged-in user.

To perform CURD OPERATIONS

l) Download Codelgniter

2) Make Database and Configuration

3) Create Routes

4) Add Item CRUD Controller

5) create Item CRUD Model

6) Create View Files

Practical 8

To install and setup, configure the Laravel Framework

Before getting started with Laravel installation, make sure your server meets the following requirements.
(ii) PHP>=5.6.4
(iii).OpenSSL PHP Extension
(iv).PDO PHP Extension
(v) Mbstring PHP Extension
(vi).Tokenizer PHP Extension
(vii)XML PHP Extension

Laravel uses Composer to manage dependencies. So before installing Laravel, make sure that the composer is installed in your system.

l. Install Composer on Windows

- Download the Composer-Setup.exe file on your system and run the installer. It will install the latest version of Composer.
- Once the Composer is installed, open the command prompt and type composer command to check the installation status.

2. Install Laravel on Windows

Assume that you want to install Laravel on the XAMP P server in Windows.

- Open the Windows command prompt and navigate to the C:\xampp\htdocs directory.
- Run the following command to install Laravel in C:\xampp\htdocs\laravel-app directory.

```
composer create-project laravel/laravel laravel-app
```

Laravel installation will start, wait for completion

Practical 9

To construct the any simple web application using Laravel Framework

Steps to create your first Laravel Application using Laravel tutorial

l) Create Your Project:

If you didn't create your project in installation section, create now by executing below command:

```
laravel new todo
```

2) Configure Database:

We need database for our application, so it's best to configure our database before doing anything. Laravel supports following 4 databases _

(i). MySQL
(ii)Postgres
(iii)SQLite
(iv)SQL Server

3) Make Auth
4) Migrations
5) Eloquent Models
6) One-to-Many Relationship
7) Artisan Tinker - (Totally Optional)
8) Controllers
9) Routing
10) Views — Blade Templates
I l) Route-Model Binding
12) Editing views
13) Run the project in Localhost