|  |  |  |  |
| --- | --- | --- | --- |
| logo-CvSU-2  Republic of the Philippines  **CAVITE STATE UNIVERSITY**  **Bacoor City Campus**  SHIV, Molino VI, City of Bacoor  **DEPARTMENT OF COMPUTER STUDIES**  **ITEC 106: IT Elective 2 (Web Systems and Technologies 2)**  **Activity Sheet**  **2nd Semester, A.Y. 2021-2022** INTENDED LEARNING OUTCOMES   **After the completion of the unit, students will be able to:**   1. Understand the importance and differences between cookies and sessions. 2. Write a PHP program that supports cookies, sessions and sends an email. | | | |
| **activity 04: CONTACT FORM** | | | |
|  | | | |
| DIRECTION  * Write a PHP program that enables the user to send an email through a contact form. * Design the contact form in HTML and CSS. * Attach your activity worksheet and activity files here. | | | |
| **rubrics for grading:** | | |
|  | | |
| **Category** | **Description** | |
| **Program**  **Execution (25pts)** | Program executes correctly with no syntax or runtime errors. | |
| **Correctness**  **(25pts)** | Program produces correct answers or appropriate results for all  inputs tested. | |
| **Layout**  **(25pts)** | The page has an exceptionally attractive and usable layout. It is  easy to locate all important elements. White space, graphic  elements and/or alignment are used effectively to organize  material. | |
| **Logic**  **(25pts)** | Program logic is correct, with no known boundary errors, and  no redundant or contradictory conditions. | |



Republic of the Philippines

**CAVITE STATE UNIVERSITY**

**Bacoor City Campus**

SHIV, Molino VI, City of Bacoor

**DEPARTMENT OF COMPUTER STUDIES**

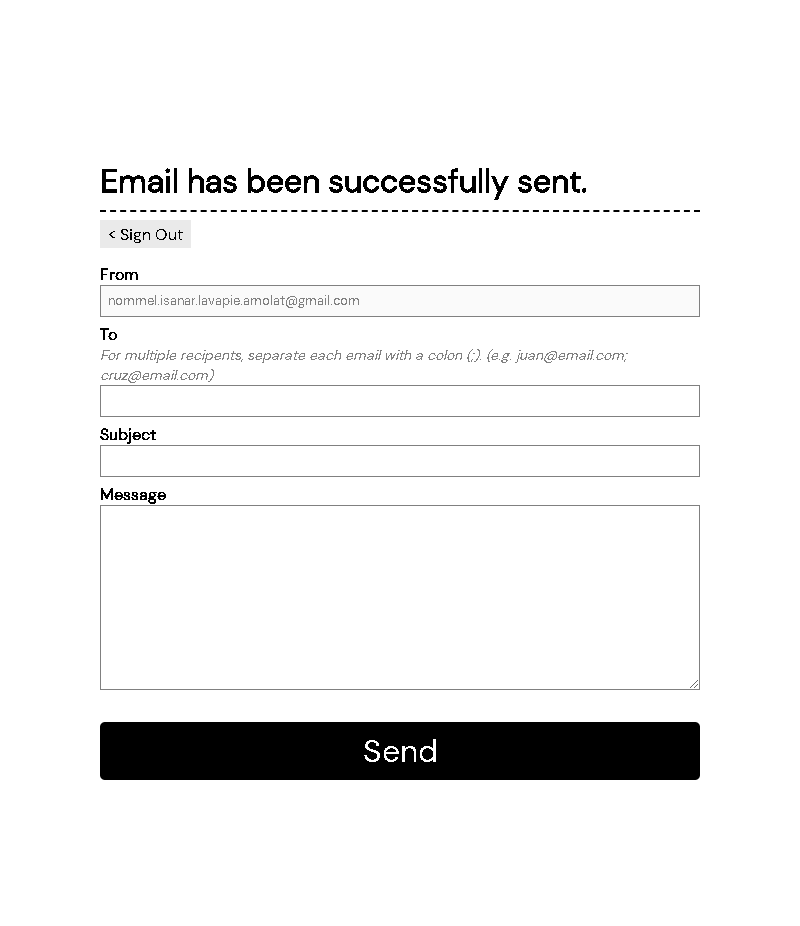
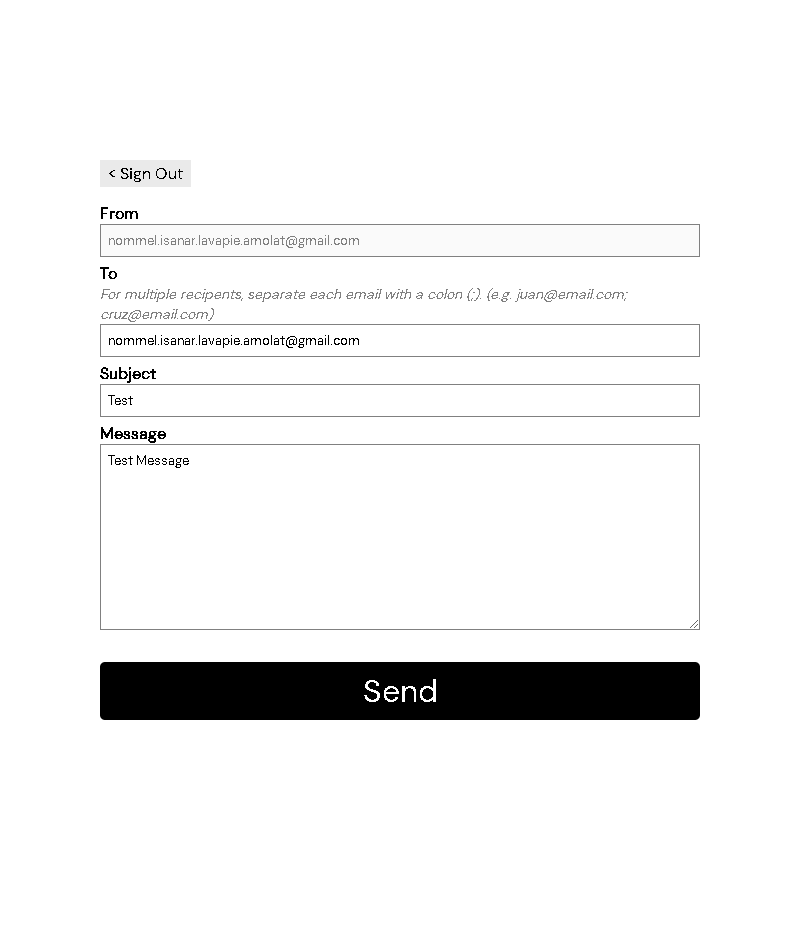
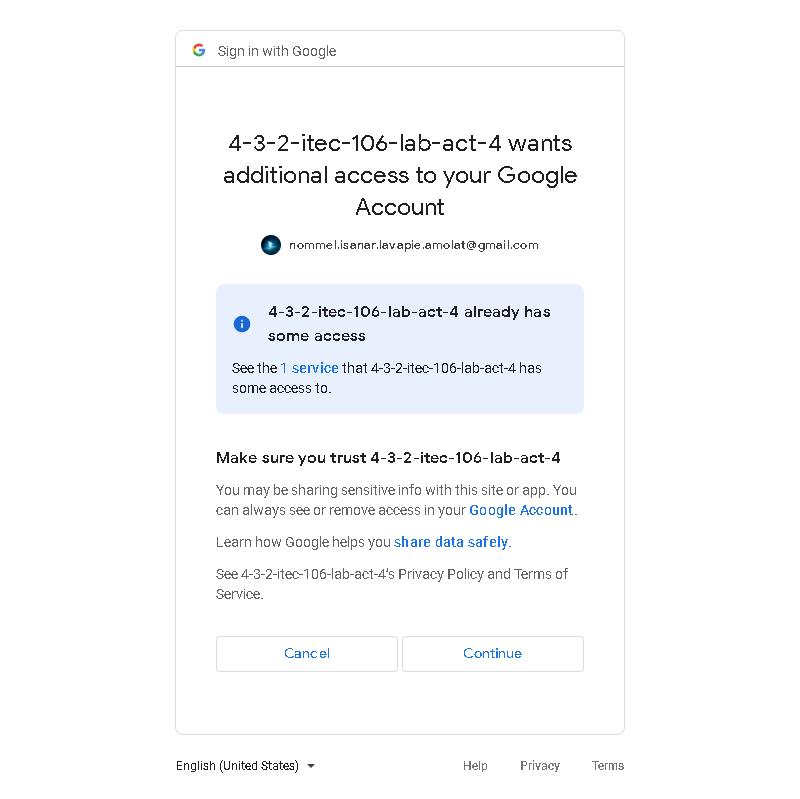
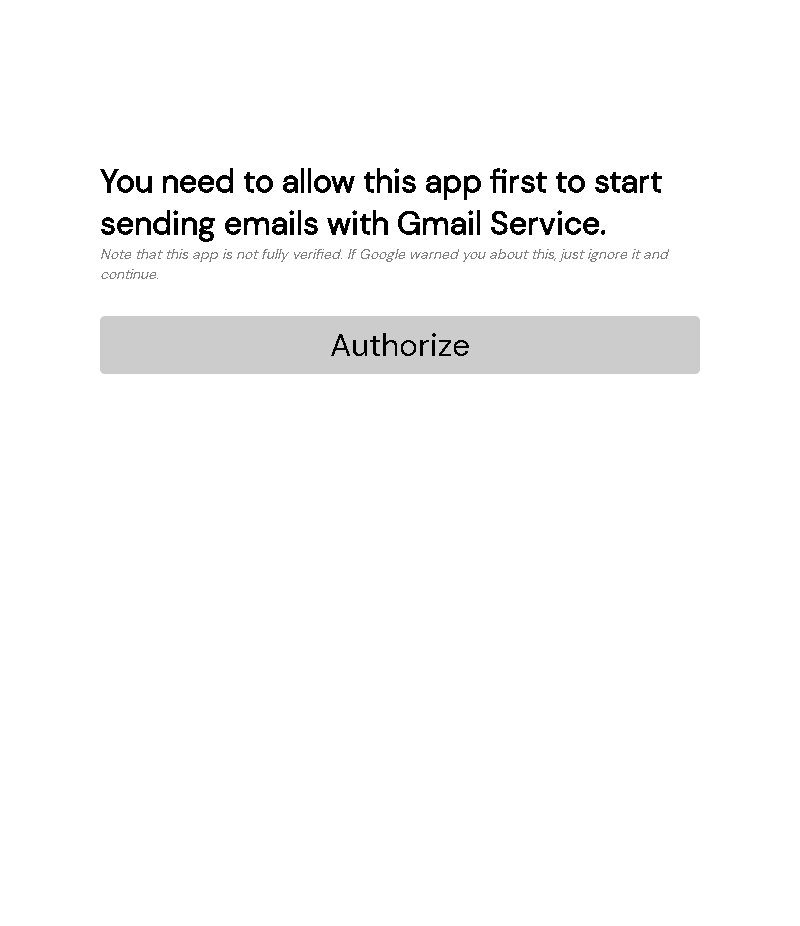
**ITEC 106: IT Elective 2 (Web Systems and Technologies 2)**

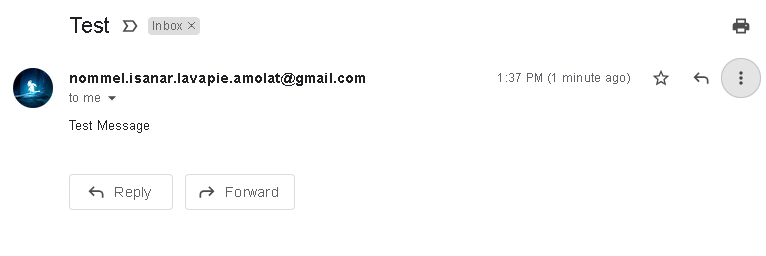
**Activity Sheet**

**2nd Semester, A.Y. 2021-2022**

|  |  |
| --- | --- |
| **Name: Nommel Isanar L. Amolat** | **Date: April 29, 2022** |
| **Course/Yr&Section: BSIT 4-3** | **Score:** |

|  |
| --- |
| **screenshot** |
|  |





|  |
| --- |
| **source code** |
|  |

<!-- index.php -->

<?php

    require\_once("./vendor/autoload.php");

    require\_once("./client.php");

    require\_once("./mailer.php");

    if(isset($\_REQUEST["sign-out"])) {

        if(deleteToken()) header("Location: ./");

    }

    $sent = null;

    if(!$authUrl && isset($\_REQUEST["submit"])) {

        $email = $\_REQUEST["email"];

        $sent = sendEmail($email["to"], $email["subject"], $email["message"]);

    }

    if(!$authUrl) {

        $authenticated = authenticated();

    }

?>

<!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">

    <link rel="stylesheet" href="style.css">

    <title>BSIT 4-3 | ITEC 106 | Laboratory Activity 3 | PHP File Manipulation</title>

</head>

<body>

    <div id="main-content">

        <?php if($sent != null) { ?>

            <?php if($sent) { ?>

                <h1>Email has been successfully sent.</h1>

            <?php } else { ?>

                <h1>Something went wrong while sending your email. Make sure that you're following the correct format.</h1>

            <?php } ?>

            <br>

        <?php } ?>

        <?php if(!$authUrl) { ?>

            <a id="sign-out" href="?sign-out">< Sign Out</a>

            <form action="./" method="post">

                <div class="field">

                    <label for="email\_from">From</label>

                    <input type="email" id="email\_from" value="<?= $authenticated ? $authenticated["emailAddress"] : "" ?>" readonly disabled>

                </div>

                <div class="field">

                    <label for="email\_to">To</label>

                    <i class="hint">For multiple recipents, separate each email with a colon (;). (e.g. juan@email.com; cruz@email.com)</i>

                    <input type="email" id="email\_to" name="email[to]" required>

                </div>

                <div class="field">

                    <label for="email\_subject">Subject</label>

                    <input type="text" id="email\_subject" name="email[subject]">

                </div>

                <div class="field">

                    <label for="email\_message">Message</label>

                    <textarea name="email[message]" id="email\_message" cols="30" rows="10"></textarea required>

                </div>

                <button type="submit" name="submit" value="1" class="submit">Send</button>

            </form>

        <?php } else { ?>

            <h1>You need to allow this app first to start sending emails with Gmail Service.</h1>

            <i class="hint">Note that this app is not fully verified. If Google warned you about this, just ignore it and continue.</i>

            <a href="<?= $authUrl ?>" class="submit">Authorize</a>

        <?php } ?>

    </div>

</body>

</html>

<!-- client.php -->

<?php

    require\_once("./vendor/autoload.php");

    $dotenv = Dotenv\Dotenv::createImmutable(\_\_DIR\_\_);

    $dotenv->load();

    use Google\Client;

    use Google\Service\Gmail;

    $accessTokenPath = './token.json';

    function loadToken() {

        global $accessTokenPath;

        if(file\_exists($accessTokenPath)) {

            return json\_decode(file\_get\_contents($accessTokenPath), true);

        }

        return null;

    }

    function saveToken($token) {

        if($token == null) return;

        global $accessTokenPath;

        if (!file\_exists(dirname($accessTokenPath))) {

            mkdir(dirname($accessTokenPath), 0700, true);

        }

        file\_put\_contents($accessTokenPath, json\_encode($token));

    }

    function deleteToken() {

        global $accessTokenPath, $authUrl;

        if(!file\_exists($accessTokenPath)) return false;

        $res = unlink($accessTokenPath);

        $authUrl = null;

        return $res;

    }

    function setAccessToken($token) {

        global $client;

        $client->setAccessToken($token);

        saveToken($token);

    }

    function authenticated() {

        global $client;

        if($client->isAccessTokenExpired()) return null;

        try {

            global $mailer;

            return $mailer->users->getProfile("me");

        } catch (Throwable $e) {

            return null;

        }

    }

    function authenticate() {

        global $client;

        $accessToken = loadToken();

        if($accessToken != null) setAccessToken($accessToken);

        else {

            if(isset($\_GET["code"])) {

                $accessToken = $client->fetchAccessTokenWithAuthCode($\_GET["code"]);

                // Check to see if there was an error.

                if (array\_key\_exists('error', $accessToken)) {

                    return $client->createAuthUrl();

                }

                setAccessToken($accessToken);

                header("Location: ./index.php");

            }

        }

        if ($client->isAccessTokenExpired()) {

            $refreshToken = $client->getRefreshToken();

            if ($refreshToken)

                setAccessToken(

                    $client->fetchAccessTokenWithRefreshToken($refreshToken));

            else return $client->createAuthUrl();

        }

        return null;

    }

    $client = new Client();

    // $client->setAuthConfig('./credentials.json');

    $client->setAuthConfig(array(

        "web" => array(

            "client\_id" => $\_ENV["WEB\_CLIENT\_ID"],

            "project\_id" => $\_ENV["WEB\_PROJECT\_ID"],

            "auth\_uri" => $\_ENV["WEB\_AUTH\_URI"],

            "token\_uri" => $\_ENV["WEB\_TOKEN\_URI"],

            "auth\_provider\_x509\_cert\_url" => $\_ENV["WEB\_AUTH\_PROVIDER\_X509\_CERT\_URL"],

            "client\_secret" => $\_ENV["WEB\_CLIENT\_SECRET"],

            "redirect\_uris" => [

                $\_ENV["WEB\_REDIRECT\_URIS\_0"]

            ]

        )

    ));

    $client->setScopes(Gmail::GMAIL\_COMPOSE, Gmail::GMAIL\_READONLY);

    $client->setAccessType('offline');

    $client->setPrompt('select\_account consent');

    $authUrl = authenticate();

    $mailer = $authUrl ? null : new Gmail($client);

?>

<!-- mailer.php -->

<?php

    require\_once("./vendor/autoload.php");

    require\_once("./client.php");

    use Google\Service\Gmail\Message;

    use Symfony\Component\Mime\Email;

    function sendEmail($to, $subject, $message) {

        global $mailer;

        if($mailer == null) return false;

        if(!is\_array($to)) $to = explode(";", $to);

        try {

            $email = (new Email())

                ->from(authenticated()["emailAddress"])

                ->to(...$to)

                ->subject($subject)

                ->text($message);

            $message = new Message();

            $message->setRaw(base64\_encode($email->toString()));

            $mailer->users\_messages->send("me", $message);

            return true;

        } catch (\Throwable $th) {}

        return false;

    }

?>

/\* style.css \*/

@import "https://fonts.googleapis.com/css2?family=DM+Mono&family=DM+Sans&display=swap";

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    /\* font-size: 11pt; \*/

    font-family: 'DM Sans', sans-serif;

}

body {

    /\* text-align: center; \*/

}

#main-content {

    width: 75%;

    max-width: 640px;

    margin-top: 10em;

    margin-bottom: 10em;

    /\* border: 1px solid red; \*/

    margin-left: auto;

    margin-right: auto;

}

#foods {

    background: rgba(128, 128, 128, 0.07);

    margin: 1em;

    padding: 0.5em;

}

li {

    margin-left: 2em;

}

.hint {

    display: block;

    color: grey;

    font-size: 0.9em;

}

.submit {

    display: block;

    width: 100%;

    margin-top: 1em;

    font-size: 2em;

    color: white;

    background: black;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    text-align: center;

    text-decoration: none;

    padding: .25em;

}

.submit:hover {

    color: black;

    background: rgba(0, 0, 0, 0.199);

}

br {

    display: block;

    content: "";

    border-bottom: 0.15em dashed black;

    margin-top: 0.5em;

    margin-bottom: 0.5em;

}

.field > label {

    display: block;

    font-weight: bolder;

}

.field > input, .field > textarea {

    display: block;

    width: 100%;

    border: 1px solid grey;

    padding: 0.5em;

    margin-bottom: 0.5em;

}

.field > input[readonly], .field > textarea[readonly] {

    color: gray;

}

#sign-out {

    text-decoration: none;

    display: inline-block;

    background-color: rgba(128, 128, 128, 0.164);

    color: black;

    padding: 0.2em;

    padding-left: 0.5em;

    padding-right: 0.5em;

    margin-bottom: 1em;

}

#sign-out:hover {

    text-decoration: 2px underline black;

}