Aim: Using regular expressions check for the validity of entered email-id. The @ symbol should not appear more than once. The dot (.) can appear at the most once before @ and at the most twice or at least once after @ symbol. The substring before @ should not begin with a digit or underscore or dot or @ or any other special character. (Use explode and ereg function.)

Methodology followed:

```html
<!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>Validation</title>

</head>

<body>
    <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]) ?>" method="POST">
        <div>
            <span>
                Email ID:
            </span>
            <input type="text" name="email" id="email" />
        </div>

        <input type="submit" value="Check" />
    </form>


    <?php
    function validateEmail($email)
    {
        // Check if @ symbol appears only once
        if (substr_count($email, '@') !== 1) {
            return false;
        }

        // Split the email address into two parts: local and domain
        list($local, $domain) = explode('@', $email);

        // Check substring before @ symbol
        if (preg_match('/^[\d.@]|[^a-zA-Z0-9.-]/', $local)) {
            return false;
        }

        // Check the domain part
        if (!preg_match('/^[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/', $domain)) {
```

```php
            return false;
        }

        // Check the position of dot (.)
        $dotPosition = strpos($local, '.');
        if ($dotPosition !== false && $dotPosition > strpos($local, '@')) {
            return false;
        }

        return true;
    }

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (empty($_POST["email"])) {
            echo "<h1>All fields are required</h1>";
            return;
        }
        $email = $_POST["email"];


        echo "<h3>";
        if (validateEmail($email)) {
            echo "$email is a valid email address.\n";
        } else {
            echo "$email is not a valid email address.\n";
        }
        echo "</h3>";
    }


    ?>

</body>

</html>
```
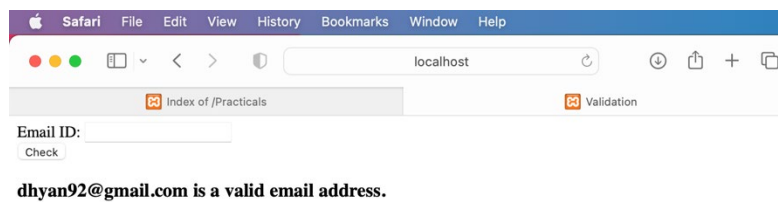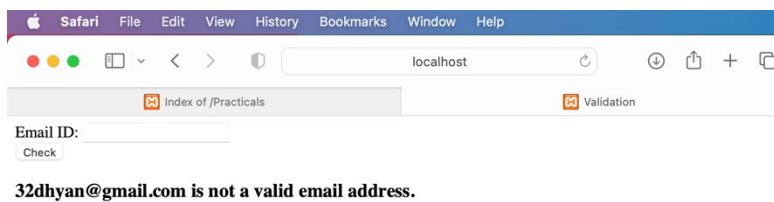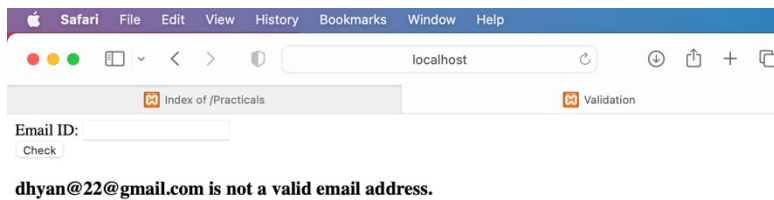
Output:

Email ID:
Check

dhyan92@gmail.com is a valid email address.

Safari File Edit View History Bookmarks Window Help

localhost

Index of /Practicals   Validation

Email ID:
Check

dhyan@22@gmail.com is not a valid email address.

Safari File Edit View History Bookmarks Window Help

localhost

Index of /Practicals   Validation

Email ID:
Check

32dhyan@gmail.com is not a valid email address.

Practical No : 9B
Aim: Using regular expressions validate mobile number and password.

Methodology Followed:

```html
<!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>Mobile Validation</title>


</head>

<body>
    <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]) ?>" method="POST">
        <div>
```

```html
        <span>
            Mobile No:
        </span>
        <input type="tel" name="mobile" id="mobile" />
    </div>
    <div>
        <span>
            Password:
        </span>
        <input type="tel" name="password" id="password" />
    </div>
    <input type="submit" value="Check" />
</form>
```
```php
<?php
function validateMobileNumber($number)
{
    $pattern = '/^\d{10}/';
    return preg_match($pattern, $number);
}


function validatePassword($password)
{
    $pattern = '/^(?=.[a-z])(?=.[A-Z])(?=.\d)(?=.[^a-zA-Z\d\s:]).{8,}$/';
    return preg_match($pattern, $password);
}


if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["mobile"]) && empty($_POST["password"])) {
        echo "<h1>All fields are required</h1>";
        return;
    }
    $mobileNumber = $_POST["mobile"];
    $password = $_POST["password"];


    echo "<h3>";
    if (validateMobileNumber($mobileNumber)) {
        echo "$mobileNumber is a valid mobile number.\n";
    } else {
        echo "$mobileNumber is not a valid mobile number.\n";
    }

    if (validatePassword($password)) {
        echo "$password is a valid password.\n";
    } else {
        echo "$password is not a valid password.\n";
    }
    echo "</h3>";
}
```
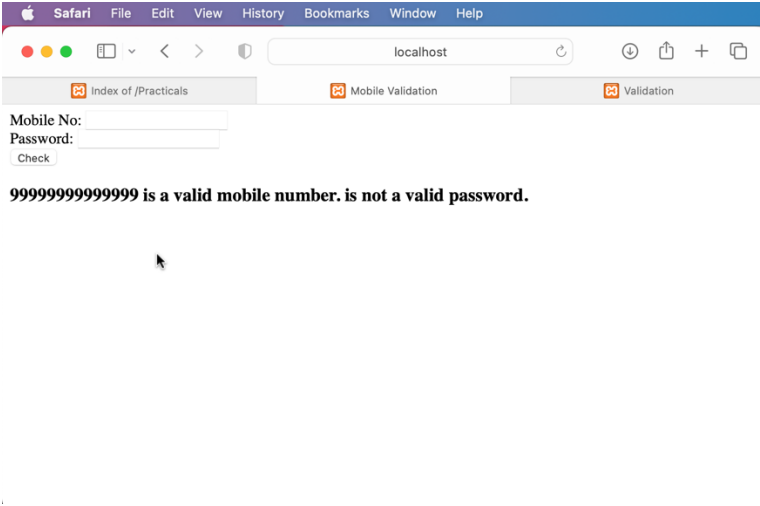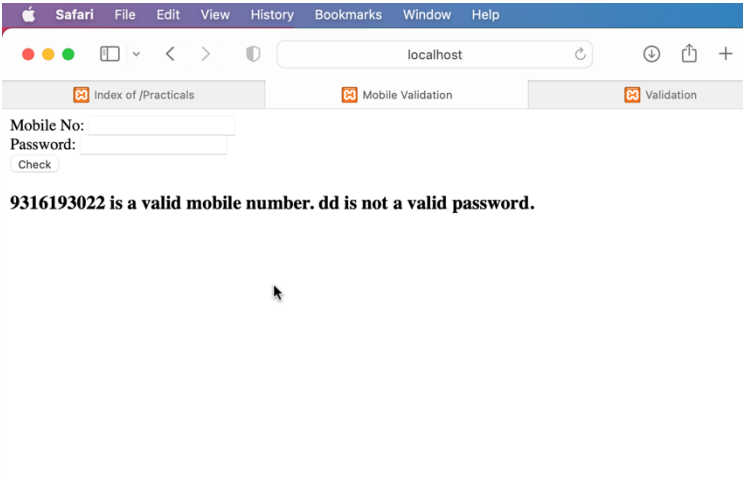
```
    ?>

</body>

</html>
```

9316193022 is a valid mobile number. dd is not a valid password.

99999999999999 is a valid mobile number. is not a valid password.

Conclusion : We learned about validating email , mobile number and password using regex.