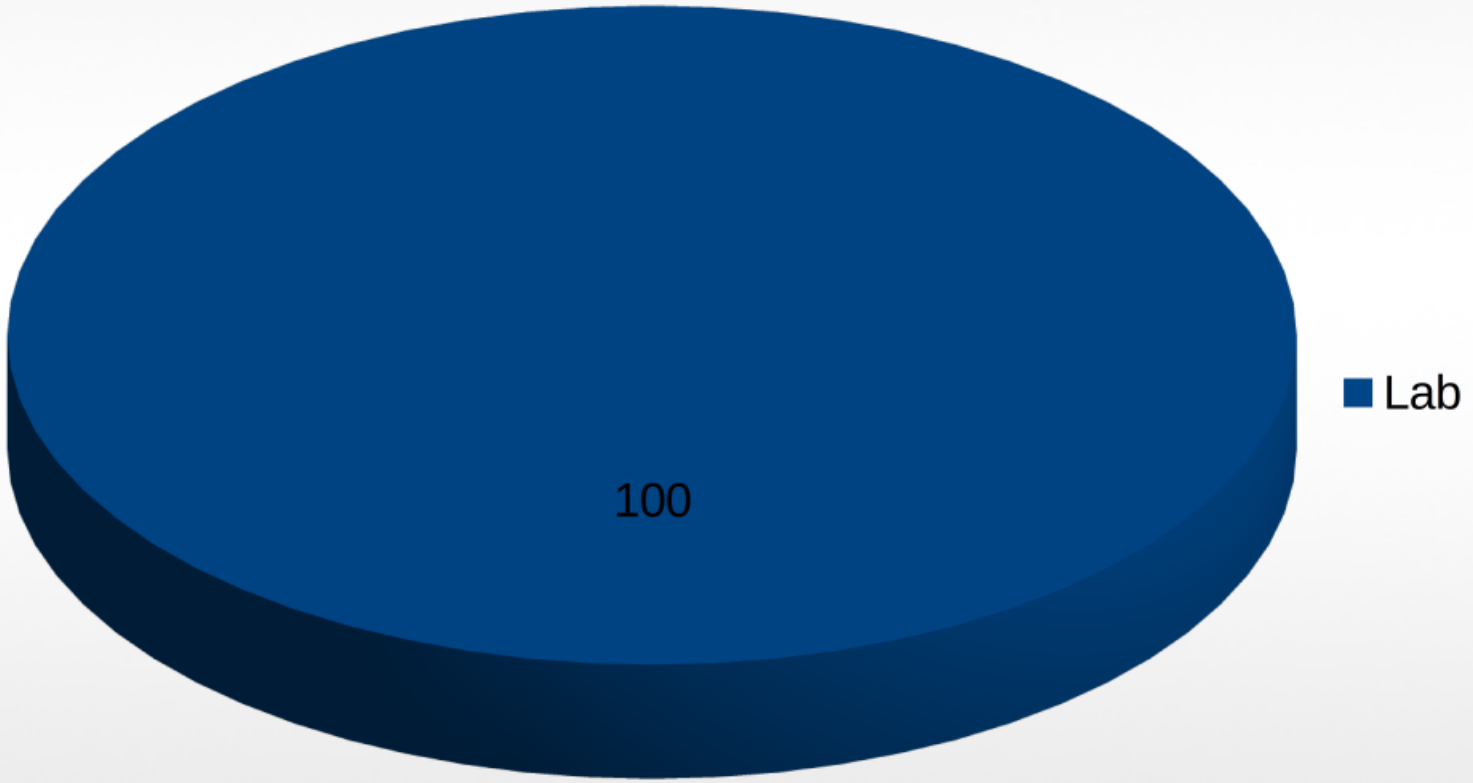
Push & Pull Tech.

AJAX in Action

Course Objectives

- Learn how to use AJAX and reversed AJAX with your web applications

Course Evaluation



Course Prerequisites

- HTML
- Javascript
- XML
- Any server side programming language

  (i.e. PHP, Python with django)

Course Prerequisites

- HTML
- Javascript
- XML
- Any server side programming language

  (i.e. PHP, Python with django)

Agenda

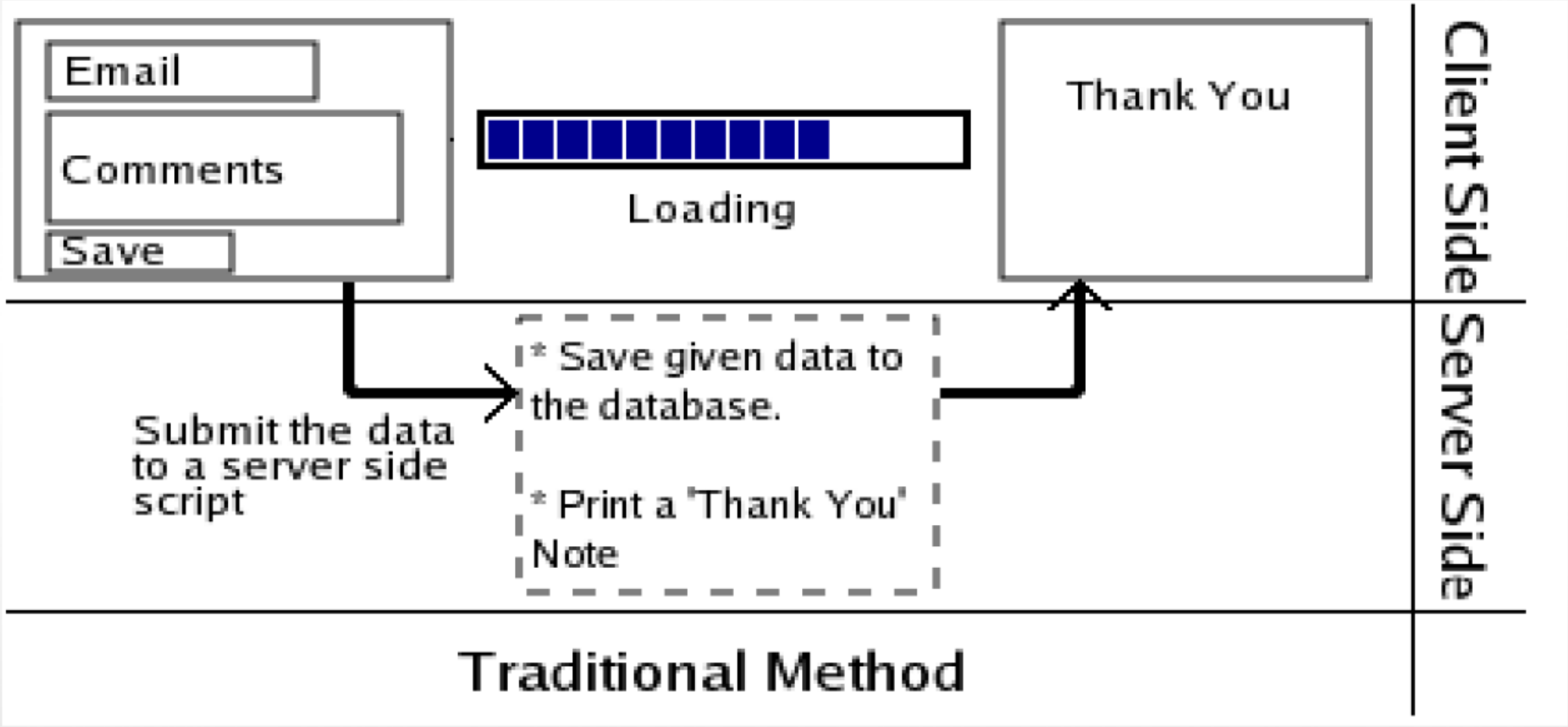
- Introduction
- What is AJAX ?
- Why AJAX?
- Getting Started
- AJAX using raw strings
- AJAX using JSON
- AJAX using XML

Introduction

Introduction

- When javascript was released, people loved all the cool things you could do with the web browser to make a more user-friendly experience.
- If you wanted to get any information from a database on the server, or send user information to a server-side script like PHP, you had to make an HTML form to GET or POST data to the server.
- The user would then have to click "Submit", wait for the server to respond, then a new page would load with the results.

Introduction



What is AJAX

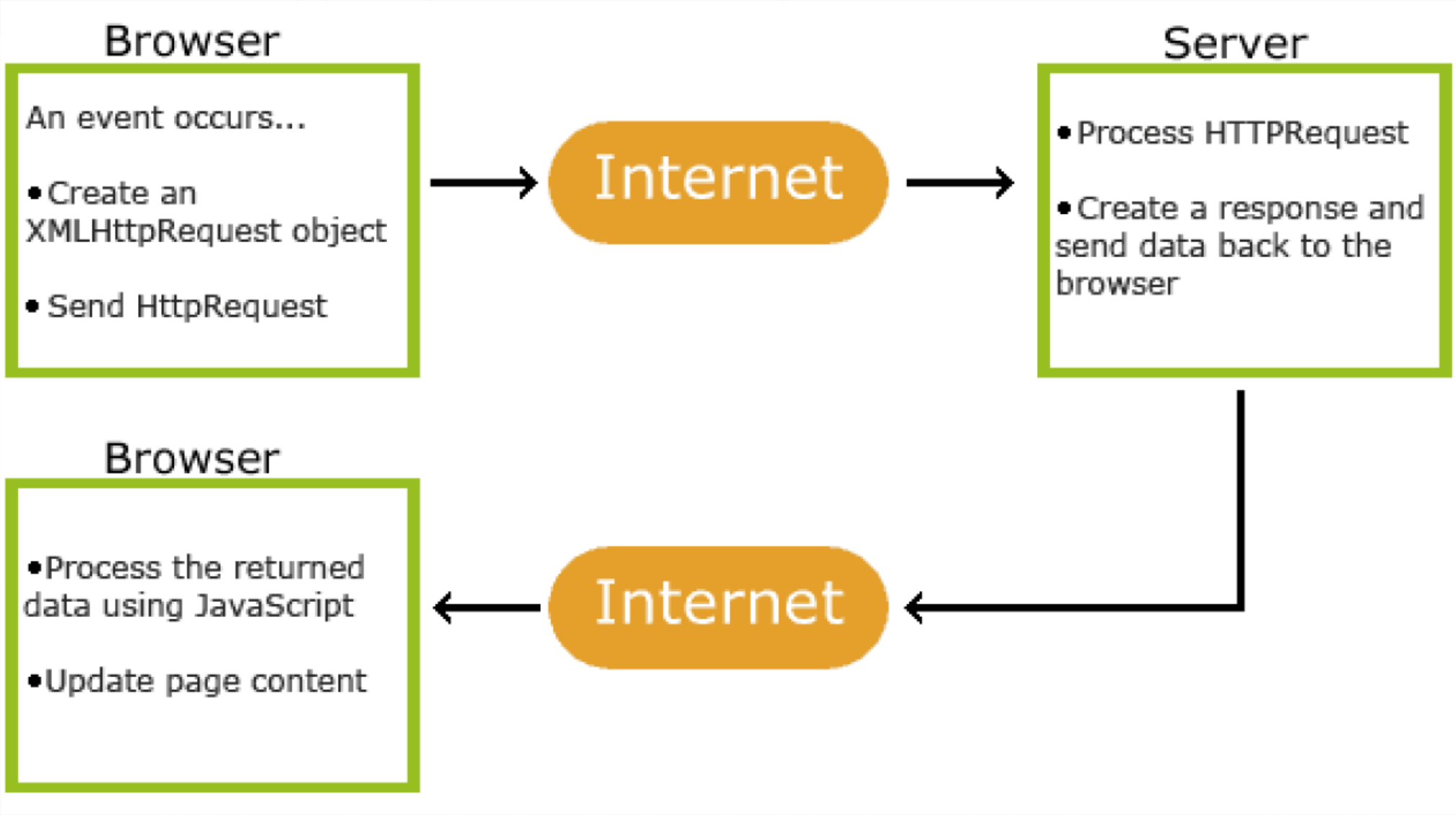
What is AJAX

Ajax: Asynchronous JavaScript and XML

- It is not a programming language.
- It is a particular way of using JavaScript.
- Enables the programmer to execute a server-side script without refreshing the page.
- It downloads data from a server in the background allows dynamically updating a page without making the user wait.
- avoids the "click-wait-refresh" pattern

- Example: Google Suggest

What is AJAX


Browser
An event occurs...
• Create an XMLHttpRequest object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Browser
• Process the returned data using JavaScript
• Update page content
Internet

Why AJAX?

Why AJAX?

- **AJAX** is based on internet standards, and uses a combination of:
  - XMLHttpRequest object (to exchange data asynchronously with a server)
  - JavaScript/DOM (to display/interact with the information)
  - CSS (to style the data)
  - XML (often used as the format for transferring data)
- **AJAX** applications are browser- and platform-independent
- **AJAX** can create a very dynamic web interface

Why AJAX?

- Your page will be more pleasant to use, when you can update parts of it without a refresh, which causes the browser to flicker and the statusbar to run.
- Because you only load the data you need to update the page, instead of refreshing the entire page, you save bandwidth.

Getting Started

Getting Started

- The first and most important piece of the Ajax puzzle was the **XMLHttpRequest (XHR)** API.
- **XHR** is a JavaScript API used to transfer data messages between a web browser and a web server.
- It allows the browser to use an **HTTP POST** (to pass data to the server) or **GET** request (to access data from the server behind the scenes)
- This API is the core of most Ajax interactions and one of the foundation technologies of modern web development.

Getting Started

- All modern browsers support the **XHR** object (IE5 and IE6 use an ActiveXObject).

```
if (window.XMLHttpRequest) {
    // code for IE7+, Firefox, Chrome, Opera, Safari
    ajaxRequest = new XMLHttpRequest();
}
else {
    // code for IE6, IE5
    ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Getting Started

Property	Description
readyState	The "readyState" property keeps track of the current stage of the request by returning an integer: 0: uninitialized,1: loading,2: loaded,3: interactive, 4: complete. ***During each stage of the request, "onreadystatechange" fires to allow you to react to it.**
status	Returns the status code of the request (integer), for example, 404 for a failed request, 200 for a successful one.
responseText	Returns the response data as a string.
responseXML	Returns the response data- assumed to be a valid XML document- as a XML object that can be easily parsed using standard DOM methods.

Getting Started

Event handler

Property	Description
onreadystatechange	Fires whenever the readyState property changes, allowing you to react to different stages of an asynchronous Ajax request.

Getting Started

Methods

Method	Description
open(method, url, [[async], [username], [password])	Sets the URL and type of the Ajax request before it is actually sent. **Method:** Enter "GET" for GET requests, "POST" for POST requests. **url:** The relative or full URL to the Ajax request. **async:** Optional Boolean parameter that specifies whether the request should be asynchronous (default) or synchronous
send(data)	Sends the request to the server with a "data" parameter specifying the body of the request. For "GET" requests, this parameter should be a value of null.

AJAX Using Raw String

Ajax Using Raw String

HTML File

```
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Using Raw Text</title>
</head>
<body>
    <input type="text" id="name"/>
    <div id="result"></div>
    <script type="text/javascript" src="js/ajax.js"></script>
</body>
</html>
```

Ajax Using Raw String

JS File

```
var nameInput = document.getElementById('name');
var result = document.getElementById('result');
nameInput.addEventListener("keyup", search);
function search() {
    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        var ajaxRequest = new XMLHttpRequest();
    }
    else {
        // code for IE6
        var ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
    }
    ajaxRequest.open("GET", "ajax.php/?name=" + nameInput.value, true);
    ajaxRequest.send();
```

Ajax Using Raw String

ajax.php

```php
<?php
$user = "root";
$password = "iti";
$host = "localhost";
$database = "iti";
$con = new mysqli($host, $user, $password, $database);
// Check connection
if ($con->connect_errno) {
    echo 'Error: Could not connect to database.';
    exit();
}
```

Ajax Using Raw String

ajax.php (cont'd)

```
$q = $_GET['name'];
$sql = 'select * from student where name like "' . $q . '%"';
$result = $con->query($sql);
for ($i = 0; $i < $result->num_rows; $i++) {
    $row = $result->fetch_assoc();
    echo $row['name'];
}
```

Ajax Using Raw String

JS File Cont'd

```
ajaxRequest.onreadystatechange = function() {
    if (ajaxRequest.readyState === 4 && ajaxRequest.status === 200) {
        result.innerHTML = '';

        //using raw text
        result.innerHTML = ajaxRequest.responseText;
         }
};
```

Ajax Using JSON

Ajax Using JSON

HTML File

```
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Using Raw Text</title>
</head>
<body>
    <input type="text" id="name"/>
    <div id="result"></div>
    <script type="text/javascript" src="js/ajax.js"></script>
</body>
</html>
```

Ajax Using JSON

JS File

```
var nameInput = document.getElementById('name');
var result = document.getElementById('result');
nameInput.addEventListener("keyup", search);
function search() {
    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        var ajaxRequest = new XMLHttpRequest();
    }
    else {
        // code for IE6
        var ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
    }
    ajaxRequest.open("GET", "ajax.php/?name=" + nameInput.value, true);
    ajaxRequest.send();
```

Ajax Using JSON

ajax.php

```php
<?php
$user = "root";
$password = "iti";
$host = "localhost";
$database = "iti";
$con = new mysqli($host, $user, $password, $database);
// Check connection
if ($con->connect_errno) {
    echo 'Error: Could not connect to database.';
    exit();
}
```

Ajax Using JSON

ajax.php (cont'd)

```
$q = $_GET['name'];
$sql = 'select * from student where name like "' . $q . '%"';
$result = $con->query($sql);
//using JSON
for ($i = 0; $i < $result->num_rows; $i++) {
    $row = $result->fetch_assoc();
    $names[$i] = $row['name'];
}
$json['names']= $names;
echo json_encode($json);
mysqli_close($con);
```

Ajax Using JSON

JS File Cont'd

```
ajaxRequest.onreadystatechange = function() {
    if (ajaxRequest.readyState === 4 && ajaxRequest.status === 200) {
        result.innerHTML = '';
        //using json
        var response = JSON.parse(ajaxRequest.responseText);
        for (i in response['names'])
            result.innerHTML += response['names'][i] + '<br/>';
        }
};
```

Ajax Using XML

Ajax Using XML

HTML File

```
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Using Raw Text</title>
</head>
<body>
    <input type="text" id="name"/>
    <div id="result"></div>
    <script type="text/javascript" src="js/ajax.js"></script>
</body>
</html>
```

Ajax Using XML

JS File

```
var nameInput = document.getElementById('name');
var result = document.getElementById('result');
nameInput.addEventListener("keyup", search);
function search() {
    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        var ajaxRequest = new XMLHttpRequest();
    }
    else {
        // code for IE6
        var ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
    }
    ajaxRequest.open("GET", "ajax.php/?name=" + nameInput.value, true);
    ajaxRequest.send();
```

Ajax Using XML

ajax.php

```php
<?php
$user = "root";
$password = "iti";
$host = "localhost";
$database = "iti";
$con = new mysqli($host, $user, $password, $database);
// Check connection
if ($con->connect_errno) {
    echo 'Error: Could not connect to database.';
    exit();
}
```

Ajax Using XML

ajax.php (cont'd)

```
$q = $_GET['name'];
$sql = 'select * from student where name like "' . $q . '%"';
$result = $con->query($sql);
//using XML
$xml = new SimpleXMLElement('<xml></xml>');
for ($i = 0; $i < $result->num_rows; $i++) {
    $row = $result->fetch_assoc();
    $student = $xml->addChild('student');
    $student->addChild('name', $row['name']);
    $student->addChild('age', $row['age']);
}
echo $xml->asXML();
mysqli_close($con);
```

Ajax Using XML

JS File Cont'd

```
ajaxRequest.onreadystatechange = function() {
    if (ajaxRequest.readyState === 4 && ajaxRequest.status === 200) {
        result.innerHTML = '';
        //          using XML
        var response = ajaxRequest.responseXML;
        var students = response.getElementsByTagName('student')
        for(var i in students)
            result.innerHTML += students[i].firstChild.firstChild.nodeValue + '</br>';
    }
};
```

Lab

1) Search Box
2) Ajax Registration Form.
3) Grid View for registries with Add ,Edit and Delete Actions using Ajax.