1.restaurant.html

```
<!DOCTYPE html>
<html>
<head>
<title>Restaurant Combos</title>
<style>
 img { width: 500px; }
 #info { margin-top: 15px; font-weight: bold; }
</style>
</head>
<body>
<h2>Restaurant – Combo Offers</h2>
<!-- Sample food image -->
<img src="https://i.imgur.com/4AiXzf8.jpeg" usemap="#foods">
<map name="foods">
 <!-- Demo hotspots (adjust coords as needed) -->
 <area shape="rect" coords="10,10,200,200" href="#"
     onclick="show('Veg Combo','₹199'); return false;">
 <area shape="rect" coords="210,10,400,200" href="#"
     onclick="show('Non-Veg Combo','₹249'); return false;">
</map>
<div id="info">Click on a hotspot to see details</div>
<script>
function show(name, price) {
 document.getElementById("info").innerHTML =
   "Combo: " + name + " — Price: " + price;
}
</script>
</body>
</html>
```

1.college.html

```
<!DOCTYPE html>
<html>
<head>
<title>College Website</title>
<!-- Embedded CSS -->
<style>
  .embedded-box {
    background: #e0f7fa;
    padding: 10px;
    border: 1px solid #0097a7;
  }
</style>
<!-- External CSS file -->
<link rel="stylesheet" href="college.css">
</head>
<body>
<h2>My College Website</h2>
<!-- Inline CSS -->
<div style="background:#fff3cd; padding:10px;">
  <h3>Inline CSS Example</h3>
  <p>This section uses inline styles.</p>
</div>
<!-- Embedded CSS -->
<div class="embedded-box">
  <h3>Embedded CSS Example</h3>
  <p>This section uses CSS inside &lt;style&gt; tag.</p>
</div>
<!-- External CSS -->
<div class="external-box">
 <h3>External CSS Example</h3>
```

```
  <p>This section uses college.css file. </p>
</div>
</body>
</html>
```

college.css

```
.external-box {
  background: #e8f5e9;
  padding: 10px;
  border: 1px solid #4caf50;
}
```

2. Digital Library

Library.html

```
<!DOCTYPE html>
<html>
<body>
<h2>Digital Library Login</h2>
<form action="Library" method="post">
  <input type="hidden" name="count" value="0">
  Name: <input name="username"><br><br>
  <button>Enter</button>
</form>
</body>
</html>
```

Library.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Library extends HttpServlet {
  protected void doPost(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    String user = req.getParameter("username");
    int count = Integer.parseInt(req.getParameter("count")) + 1;
    out.println("<h2>Welcome, " + user + "</h2>");
    out.println("<p>You visited " + count + " time(s)</p>");
    out.println("<form method='post' action='Library'>");
    out.println("<input type='hidden' name='count' value='" + count + "'>");
    out.println("<input type='hidden' name='username' value='" + user + "'>");
    out.println("<button>Visit Again</button>");
    out.println("</form>");    }}
```

3. Three-Tier Application Using JSP

student.jsp

```
<%@ page import="java.util.*" %>
<html><body>

<h2>Student Marks</h2>

<%
  // Demo "Database"
  String[][] data = {
    {"101","Ravi","85"},
    {"102","Meena","91"},
    {"103","Kiran","77"}
  };
  out.println("<table border='1'>");
  out.println("<tr><th>Roll No</th><th>Name</th><th>Marks</th></tr>");
  for(String[] s : data){
    out.println("<tr>");
    out.println("<td>"+s[0]+"</td>");
    out.println("<td>"+s[1]+"</td>");
    out.println("<td>"+s[2]+"</td>");
    out.println("</tr>");
  }
  out.println("</table>");
%>
</body></html>
```

4. Indian Map With Hotspots

```
<!DOCTYPE html>
<html>
<body>
<h2>Indian States & Capitals</h2>
<img src="https://upload.wikimedia.org/wikipedia/commons/3/39/India_states_map.png"
   usemap="#india" width="450">
<map name="india">
 <area shape="rect" coords="80,250,150,330" href="#"
    onclick="show('Karnataka','Bengaluru'); return false;">
 <area shape="rect" coords="120,150,200,230" href="#"
    onclick="show('Maharashtra','Mumbai'); return false;">
</map>
<h3 id="info">Click on a State</h3>
<script>
function show(state, capital){
 document.getElementById("info").innerHTML =
   "State: " + state + " — Capital: " + capital;
}
</script>
</body>
</html>
```

5. Online Banking Registration – Client-Side Validation

```
<!DOCTYPE html>
<html>
<body>
<h2>Online Banking Registration</h2>
<form onsubmit="return validate()">
   Username: <input id="u"><br><br>
   Email: <input id="e"><br><br>
   Password: <input id="p" type="password"><br><br>
   <button>Register</button>
</form>
<p id="msg" style="color:red"></p>
<script>
function validate(){
   let u = u.value.trim();
   let e = e.value.trim();
   let p = p.value.trim();
   if(u.length < 3){ msg.innerHTML="Username too short"; return false;}
   if(!/^\S+@\S+\.\S+$/.test(e)){ msg.innerHTML="Invalid email"; return false;}
   if(p.length < 6){ msg.innerHTML="Password must be 6+ chars"; return false;}
   alert("Registration Successful!");
   return false;
}
</script>
</body>
</html>
```

shop.css

```
.external {
   background: #e8f5e9;
   padding: 10px;
   border: 1px solid #4caf50;}
```

7. University Management

login.jsp

```
<%@ page session="true" %>
<html><body>
<h2>Student Login</h2>
<form method="post">
  ID: <input name="id"><br><br>
  Password: <input type="password" name="pwd"><br><br>
  <button>Login</button>
</form>
<%
  String id = request.getParameter("id");
  String pwd = request.getParameter("pwd");
  if(id!=null){
    if(id.equals("student") && pwd.equals("123")){
      session.setAttribute("user", id);
      out.println("<h3>Login Successful</h3>");
    } else {
      out.println("<h3>Invalid Credentials</h3>");   }
  }
%>
</body></html>
```

web.xml

```
<servlet>
  <servlet-name>login</servlet-name>
  <jsp-file>/login.jsp</jsp-file>
</servlet>
<servlet-mapping>
  <servlet-name>login</servlet-name>
  <url-pattern>/login</url-pattern>
</servlet-mapping>
```

8. Airline Ticket Reservation

airline.jsp

```
<%@ page import="java.util.*" %>
<html><body>
<h2>Available Flights</h2>
<%
   String[][] flights = {
      {"AI101","Delhi → Mumbai","₹4500"},
      {"AI202","Bangalore → Kolkata","₹3200"}
   };
   for(String[] f : flights){
      out.println("<p>Flight: "+f[0]+" | Route: "+f[1]+" | Fare: "+f[2]+"</p>");
   }
%>
</body></html>
```

9. XML of Tourist Places + Query Program (HTML + JS)

tourist.html

```
<!DOCTYPE html>
<html>
<body>
<h2>Tourist Places Lookup</h2>
<input id="q" placeholder="Enter place name">
<button onclick="find()">Search</button>
<pre id="out"></pre>
<script>
const xml = `
<places>
 <place><name>Taj Mahal</name><city>Agra</city></place>
 <place><name>Charminar</name><city>Hyderabad</city></place>
 <place><name>Qutub Minar</name><city>Delhi</city></place>
</places>`;
const xmldoc = new DOMParser().parseFromString(xml, "text/xml");
function find(){
 let key = q.value.toLowerCase();
 let nodes = xmldoc.getElementsByTagName("place");
 let result = "";
 for(let p of nodes){
  let name = p.getElementsByTagName("name")[0].textContent;
  let city = p.getElementsByTagName("city")[0].textContent;
  if(name.toLowerCase().includes(key)){
   result += name+" — "+city+"\n"; } }
 out.textContent = result || "No match";
}
</script>
</body>
</html>
```

10. Web Service for College Management

Node.js server (server.js)

```
const express = require("express");
const app = express();

app.get("/departments", (req, res) => {
  res.json({
    departments: [
      {id:1, name:"CSE"},
      {id:2, name:"ECE"},
      {id:3, name:"IT"}
    ]
  });
});
app.listen(3000, () => console.log("Running on port 3000"));
```

JSON

```
{
 "departments": [
  {"id":1,"name":"CSE"},
  {"id":2,"name":"ECE"},
  {"id":3,"name":"IT"}
 ]
}
```

11. Online Supermarket – Simple Demo (HTML + JS)

supermarket.html

```
<!DOCTYPE html>
<html>
<body>
<h2>Online Supermarket</h2>
<button onclick="page('home')">Home</button>
<button onclick="page('reg')">Register</button>
<button onclick="page('cart')">Cart</button>
<div id="screen"></div>
<script>
let cart=[];
function page(p){
  if(p=="home"){
    screen.innerHTML = "<h3>Products</h3><button onclick='add()'>Add Apple</button>";
  }
  else if(p=="reg"){
    screen.innerHTML = "<h3>Registration</h3><input id='name'><button onclick='save()'>Save</button>";
  }
  else {
    screen.innerHTML = "<h3>Cart</h3>"+(cart.length?cart:"Empty");
  }
}
function add(){ cart.push("Apple"); alert("Added"); }
function save(){ alert("Registered: "+name.value); }
page("home");
</script>
</body></html>
```

12. Employee Payroll – JSP (Three Tier Demo)

payroll.jsp

```
<%@ page import="java.util.*" %>
<html><body>

<h2>Employee Payroll</h2>

<%
  List<String[]> list = Arrays.asList(
    new String[]{"201","Ravi","45000"},
    new String[]{"202","Meera","52000"}
  );

  out.println("<table border='1'>");
  out.println("<tr><th>ID</th><th>Name</th><th>Salary</th></tr>");

  for(String[] e : list){
    out.println("<tr><td>"+e[0]+"</td><td>"+e[1]+"</td><td>"+e[2]+"</td></tr>");
  }
  out.println("</table>");
%>

</body></html>
```

13. XML for Organic Stores + Query Program

organic.html

```
<!DOCTYPE html>
<html>
<body>
<h2>Organic Stores Lookup</h2>
<input id="q"><button onclick="find()">Search</button>
<pre id="out"></pre>
<script>
const xml = `
<stores>
 <store><name>Green Fresh</name><area>Main Road</area></store>
 <store><name>Nature Mart</name><area>Market</area></store>
 <store><name>Organic Hub</name><area>Station Rd</area></store>
</stores>`;
const x = new DOMParser().parseFromString(xml,"text/xml");
function find(){
 let key = q.value.toLowerCase();
 let s = x.getElementsByTagName("store");
 let out = "";
 for(let st of s){
   let n = st.getElementsByTagName("name")[0].textContent;
   let a = st.getElementsByTagName("area")[0].textContent;
   if(n.toLowerCase().includes(key)) out += n+" — "+a+"\n";
 }
 out === "" ? out="Not found" : out;
 document.getElementById("out").textContent=out;
}
</script>
</body></html>
```

14. Call Taxi Service – JSP

taxi.jsp

```
<%@ page import="java.util.*" %>
<html><body>

<h2>Call Taxi Service</h2>

<%
  String[][] taxis = {
    {"TX101","Ramesh","KA01AB1234","4km","₹120"},
    {"TX202","Suresh","KA02BC5678","6km","₹180"}
  };

  for(String[] t : taxis){
    out.println("<p>Taxi: "+t[0]+" | Driver: "+t[1]+" | Vehicle: "+t[2]+" | Distance: "+t[3]+" | Fare: "+t[4]+"</p>");
  }
%>

</body></html>
```

15. XML for E-Book Library + Search Program

ebooks.html

```
<!DOCTYPE html>
<html>
<body>
<h2>E-Book Library</h2>
<input id="q" placeholder="Book title"><button onclick="find()">Find</button>
<pre id="out"></pre>
<script>
const xml = `
<books>
 <book><title>Java Basics</title><author>Ravi</author></book>
 <book><title>Web Technology</title><author>Meera</author></book>
 <book><title>Python Guide</title><author>Kiran</author></book>
</books>`;
const doc = new DOMParser().parseFromString(xml,"text/xml");
function find(){
  let k = q.value.toLowerCase();
  let b = doc.getElementsByTagName("book");
  let r = "";
  for(let bk of b){
    let t = bk.getElementsByTagName("title")[0].textContent;
    let a = bk.getElementsByTagName("author")[0].textContent;
    if(t.toLowerCase().includes(k)) r += t+" — "+a+"\n";
  }
  out.textContent = r || "Book not found";
}
</script>
</body>
</html>
```