Application Design and Development

Ram Datta Bhatta

Content

✓User Interface & tools,

✓ Web Fundamental,

✓Servlets and JSP,

✓Authorization in SQL,

✓Application Security

Application Programs and User Interfaces

- Most database users do *not* use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces
  - Many interfaces are Web-based

User Interface

- ✓User Interface (UI) refers to the visual and interactive elements of a software application or system that enable users to interact with it.
- ✓User interface includes the design, layout, and presentation of the application's graphical user interface (GUI) components, such as buttons, menus, forms, icons, and other visual elements.
- ✓The primary goal of a user interface is to provide a user-friendly and intuitive experience, allowing users to easily navigate, interact with, and control the software or system.

UI Tools

✓UI Tools are software or frameworks that assist in the creation and development of user interfaces. These tools provide a set of features, libraries, and components to streamline the UI design and development process. Some of the tools are:

- Design Tools: Adobe XD, Sketch
- Development Frameworks and Libraries: React, Angular, Vue.js, Bootstrap
- Graphic Editors: Adobe Photoshop, Adobe Illustrator, Sketch
- IDEs and Code Editors: Visual Studio Code, Android Studio, Eclipse
- UI Animation Tools: Adobe After Effects, Principle, Framer Motion

Web Fundamental

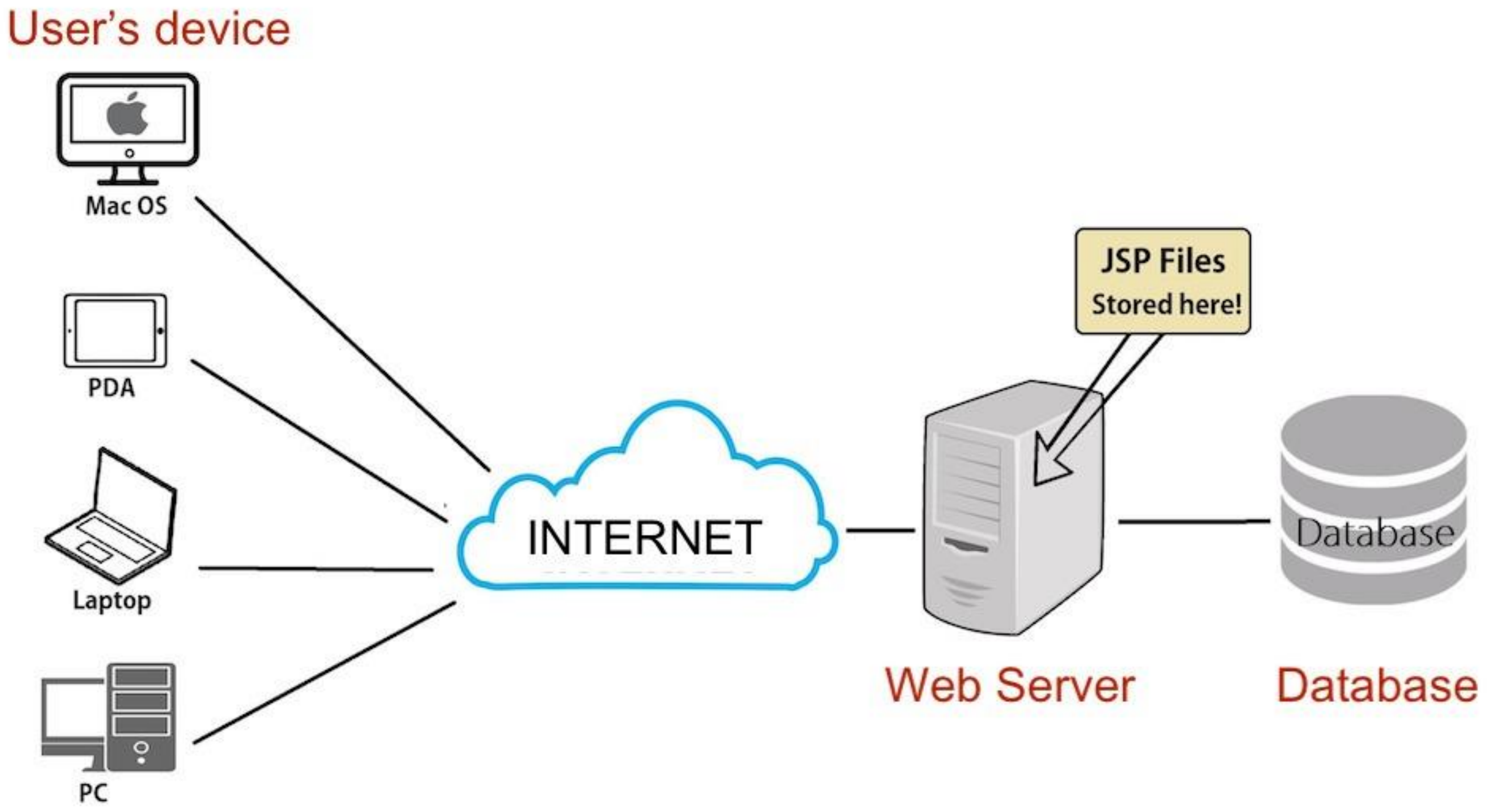
✓ The core concepts and technologies that underpin the development of websites and web applications

- HTML (Hypertext Markup Language): HTML is the standard markup language used to structure and present content on the web.
- CSS (Cascading Style Sheets): CSS is a style sheet language that controls the visual appearance of web pages. It allows developers to define the colors, fonts, layout, and other visual properties of HTML elements
- JavaScript: JavaScript is a programming language that enables interactivity and dynamic behavior in web pages. It allows developers to manipulate the HTML and CSS elements, handle user events, perform calculations, make HTTP requests, and create interactive features like animations, form validation, and responsive behavior.
- Client-Server Architecture: The web operates on a client-server model.
- HTTP (Hypertext Transfer Protocol): HTTP is the protocol used for transferring data over the web.
- Web Standards and Best Practices: Following web standards and best practices is essential for cross-browser compatibility, performance optimization, security, and maintainability.

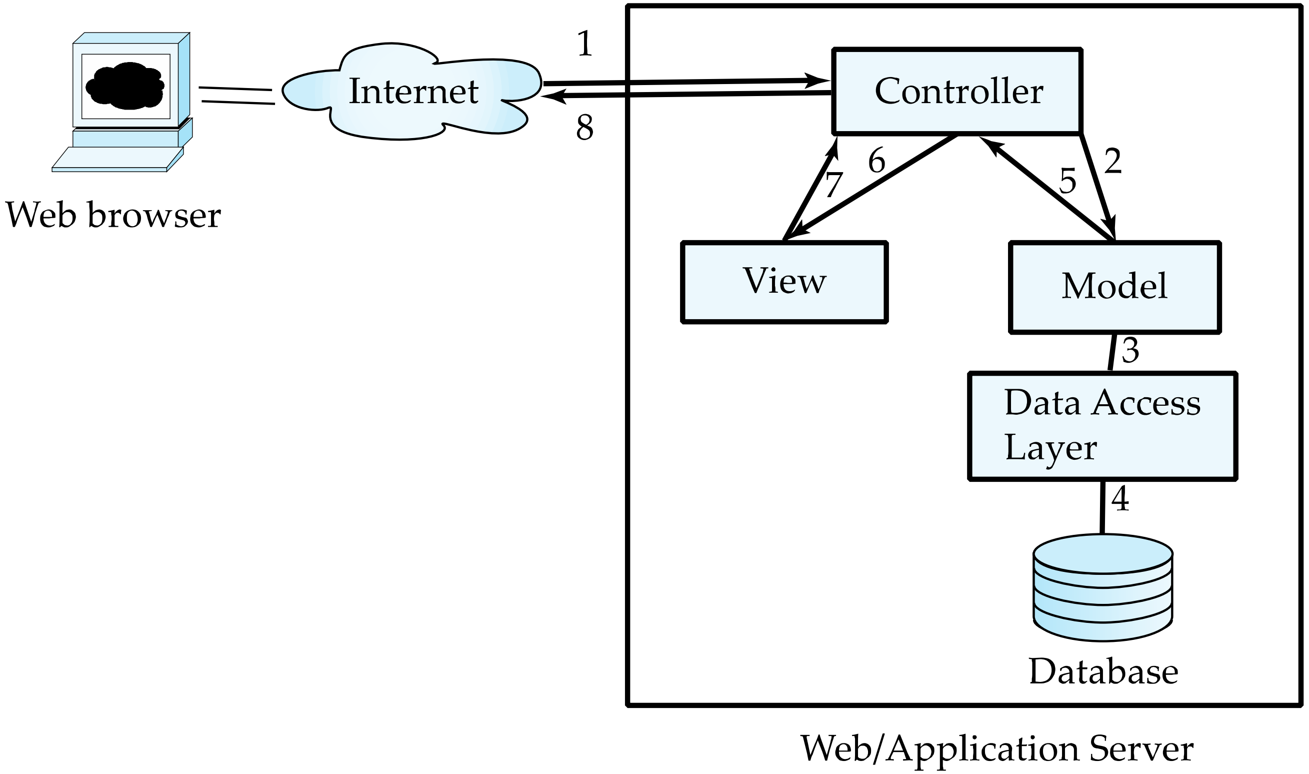
Web Interface

- ✓ Web browsers have become the de-facto standard user interface to databases
- ✓ Enable large numbers of users to access databases from anywhere
- ✓ Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
  - ▪ Javascript, Flash and other scripting languages run in browser, but are downloaded transparently
- ✓ Examples: banks, airline and rental car reservations, university course registration and grading, an so on.

Role of Database in Web application



Application Architecture



Servlets

✓Servlets are Java classes that handle the request-response cycle of a web application.

✓They receive HTTP requests from clients (web browsers or other applications) and generate dynamic responses.

✓Servlets can perform tasks such as processing form data, accessing databases, performing business logic, and generating HTML or other types of responses

Servlets

- Java Servlet specification defines an API for communication between the Web/application server and application program running in the server
    - E.g., methods to get parameter values from Web forms, and to send HTML text back to client
- Application program (also called a servlet) is loaded into the server
    - Each request spawns a new thread in the server
        - thread is closed once the request is serviced
    - Programmer creates a class that inherits from HttpServlet
        - And overrides methods doGet, doPost, …
    - Mapping from servlet name (accessible via HTTP), to the servlet class is done in a file web.xml
        - Done automatically by most IDEs when you create a Servlet using the IDE

Example Servlet Code

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class PersonQueryServlet extends HttpServlet {
  public void doGet (HttpServletRequest request, HttpServletResponse response)
                  throws ServletException, IOException
  {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<HEAD><TITLE> Query Result</TITLE></HEAD>");
    out.println("<BODY>");
      ….. BODY OF SERVLET (next slide) …
    out.println("</BODY>");
    out.close();
  }
}
```

Example Servlet Code

```
String persontype = request.getParameter("persontype");
String number = request.getParameter("name");
if(persontype.equals("student")) {
   ... code to find students with the specified name ...
   ... using JDBC to communicate with the database ..
   out.println("<table BORDER COLS=3>");
   out.println(" <tr> <td>ID</td> <td>Name: </td>" + " <td>Department</td> </tr>");
   for(... each result ...){
      ... retrieve ID, name and dept name
      ... into variables ID, name and deptname
      out.println("<tr> <td>" + ID + "</td>" + "<td>" + name + "</td>" + "<td>" + deptname
             + "</td></tr>");
   };
   out.println("</table>");
}
else {
   ... as above, but for instructors ...
}
```

Server-Side Scripting

- Server-side scripting simplifies the task of connecting a database to the Web
    - Define an HTML document with embedded executable code/SQL queries.
    - Input values from HTML forms can be used directly in the embedded code/SQL queries.
    - When the document is requested, the Web server executes the embedded code/SQL queries to generate the actual HTML document.
- Numerous server-side scripting languages
    - JSP, PHP
    - General purpose scripting languages: VBScript, Perl, Python

Java Server Pages (JSP)

- A JSP page with embedded Java code

```
<html>
<head> <title> Hello </title> </head>
<body>
<% if (request.getParameter(“name”) == null)
{ out.println(“Hello World”); }
else { out.println(“Hello, ” + request.getParameter(“name”)); }
%>
</body>
</html>
```

- JSP is compiled into Java + Servlets
- JSP allows new tags to be defined, in tag libraries
  - Such tags are like library functions, can are used for example to build rich user interfaces such as paginated display of large datasets

PHP

- PHP is widely used for Web server scripting
- Extensive libaries including for database access using ODBC

```
 <html>
 <head> <title> Hello </title> </head>
 <body>
 <?php if (!isset($_REQUEST[ ‘name’ ]))
 { echo “Hello World”; }
 else { echo “Hello, ” + $_REQUEST[ ‘name’ ]; }
 ?>
 </body>
</html>
```

JSP and Servlet

- ✓ JSP as a View Component: JSP is primarily used as a view component in a web application. It allows developers to embed Java code within HTML templates, making it easier to generate dynamic content and presentation. JSP pages are essentially compiled into Servlets behind the scenes.
- ✓ Servlets as Controller Components: Servlets handle the request-response cycle of a web application. They receive HTTP requests from clients (such as web browsers) and generate dynamic responses. Servlets contain Java code that performs the business logic, data processing, and interaction with databases or other resources.
- ✓ JSP as Servlets: When a JSP page is accessed, it is translated into a Servlet by the container (e.g., Tomcat, Jetty) that runs the web application. This translation process happens automatically during the first request or when changes are detected in the JSP file. The resulting Servlet handles the request-response cycle, executing the embedded Java code within the JSP page.

Authorization in SQL

- ✓ Authorization is the process of granting or denying permissions to users or roles for accessing and performing specific operations on database objects, such as tables, views, stored procedures, or functions. It ensures that only authorized entities can interact with the database and perform desired actions. SQL authorization involves the following elements:
- ✓ Users: Users are individual entities with unique identifiers who interact with the database.
- ✓ Roles: Roles are predefined groups or categories that represent sets of users with similar access requirements or responsibilities. Roles provide a convenient way to manage permissions for multiple users simultaneously.
- ✓ Privileges: Privileges are specific permissions granted to users or roles that define what actions they can perform on database objects. Common privileges include SELECT, INSERT, UPDATE, DELETE, EXECUTE, and others.

Examples: granting and revoking

- ✓GRANT SELECT, INSERT ON table_name TO user_or_role;
- ✓REVOKE SELECT, INSERT ON table_name FROM user_or_role;
- ✓GRANT role_name TO user;

Application Security

- ✓The practices, measures, and techniques employed to protect software applications from potential security threats and vulnerabilities.
- ✓It involves identifying and mitigating risks, ensuring the confidentiality, integrity, and availability of application data, and safeguarding against unauthorized access, misuse, or manipulation of application resources.
- ✓SQL Injection, Authentication and Authorization, Session Management, XSS, CATCHA are important for the handling of application security.

SQL Injection

- A web application security vulnerability that occurs when untrusted user-supplied data is directly inserted into a SQL query without proper validation or sanitization.
- It allows an attacker to manipulate or inject malicious SQL statements into the application's database query, potentially gaining unauthorized access to data, modifying or deleting data, or executing arbitrary commands on the database server.

✓SELECT * FROM users WHERE username = 'input_username' AND password = 'input_password';

✓SELECT * FROM users WHERE username = '' OR '1'='1' AND password = 'input_password';

Authentication and Authorization

- ✓Implement secure authentication mechanisms, such as strong password policies, multi-factor authentication, and secure session management.
- ✓ Authentication is the process of verifying the identity of a user or entity attempting to access a system or application. It ensures that users are who they claim to be.
- ✓Authorization, also known as access control, is the process of granting or denying access rights and permissions to authenticated users based on their identity, roles, and privileges. It determines what resources or actions a user is allowed to access within the application.

Secure Session Management

- ✓Ensure proper session management by using secure session cookies, setting expiration timeouts, and regenerating session identifiers upon authentication.
- ✓ Implement mechanisms to protect against session hijacking, fixation, and ensure secure session storage

Cross-Site Scripting

✓Cross-Site Scripting (XSS) is a web application security vulnerability that allows attackers to inject malicious scripts into trusted websites viewed by other users.

✓ XSS occurs when an application fails to properly validate and sanitize user-supplied input, which is then displayed to other users without appropriate encoding or escaping.

✓ This vulnerability allows attackers to execute their own scripts within the context of the affected website, potentially compromising user data or performing malicious actions on behalf of the user.

Use of CAPTCHA

- ✓CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a security mechanism used to differentiate between human users and automated bots or malicious scripts on the internet.

- ✓ CAPTCHA is a challenge-response test presented to users during certain online interactions to verify that they are human and not computer programs attempting to abuse or exploit a system.

- ✓The primary purpose of CAPTCHA is to prevent automated bots from engaging in malicious activities, such as spamming online forms, creating multiple accounts, launching brute-force attacks, or performing web scraping. CAPTCHA mechanisms aim to ensure the integrity, availability, and usability of online services by distinguishing between human users and automated scripts.

Thank you.