

# Blog App User Roles

## Blog Application with User Roles and Authentication

### Project Overview

This project aims to develop a web application for managing blog posts with user roles and authentication. The application will have two user roles: Admin and Viewer.

### Technologies

- **Backend:** Java Servlets
- **Frontend:** JSP (JavaServer Pages)
- **Database:** MySQL

### Project Phases

#### Create tables

```
CREATE TABLE posts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(255),  
    content TEXT,  
    date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    userId INT,  
    edit BOOLEAN DEFAULT false,  
    FOREIGN KEY (userId) REFERENCES user(id)  
);
```

```
CREATE TABLE media (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    postId INT,  
    file_data LONGBLOB,  
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (postId) REFERENCES posts(id) ON DELETE CASCADE  
);
```

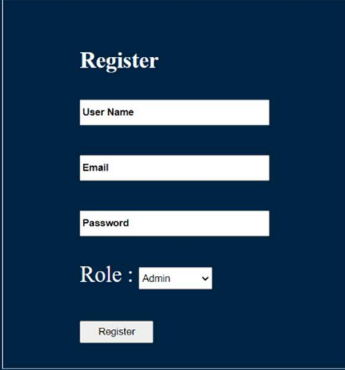
**Index.jsp**



## **Phase 1: User Registration and Login (Frontend & Backend)**

### **1. User Registration**

- Create a user registration form that captures user details (name, email, password) and role (Admin/Viewer).
- Implement user registration logic in a Servlet:
  - Validate user data.
  - Store user data in the MySQL database.
  - Assign roles (Admin/Viewer).

A screenshot of a web application's registration form. The form is titled "Register" and is set against a dark blue background. It contains four input fields: "User Name", "Email", and "Password", each with a label above it. Below these is a "Role" dropdown menu currently showing "Admin". At the bottom of the form is a "Register" button.

**Register**

User Name

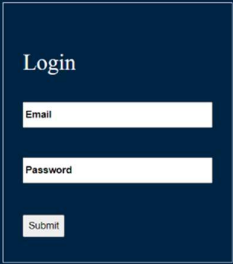
Email

Password

Role :

## 2. User Login

- Create a login page with username/email and password fields.
- Implement login logic in a Servlet:
  - Authenticate users against the database.
  - Upon successful login, establish user sessions using cookies or HTTP sessions for role identification.

A screenshot of a web application's login form. The form is titled "Login" and is set against a dark blue background. It contains two input fields: "Email" and "Password", each with a label above it. Below these is a "Submit" button.

**Login**

Email

Password

## Phase 2: Admin Dashboard (Frontend & Backend)

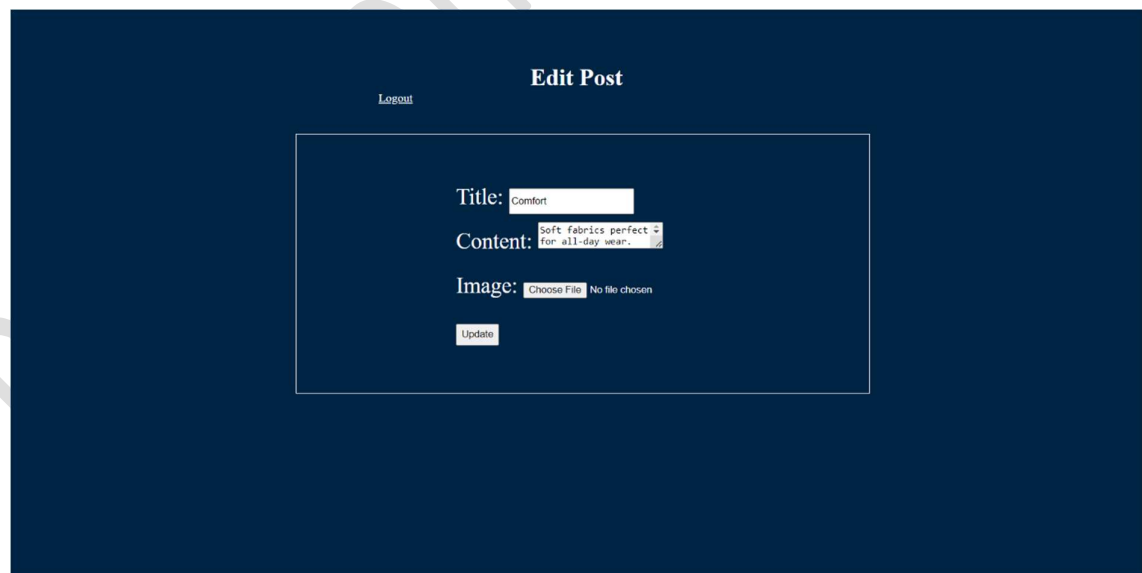
### 1. Admin Dashboard

- Develop an Admin dashboard accessible only after successful login with the "Admin" role.
- Implement functionalities for Admin to:

- Create new blog posts with title, content, and an option to upload images.
- Update existing blog posts (title, content, image).
- Delete blog posts.



Edit pOst



### Phase 3: User Interface for Viewers (Frontend)


#### 1. User Interface for Viewers

- Search blog posts by title or date.(still not implement)

- View individual blog posts in detail, opening them in a new tab.

Viewer Dashboard	
Admin	Action
John Doe	<a href="#">View</a>
ram	<a href="#">View</a>
ramesh	<a href="#">View</a>

Openend Admin John Doe in new tab

Admin Posts		
Search by Title: <input type="text"/>		
<input type="button" value="Search"/>		
Title	Post Id	
	Clarity	39
		

Video Link::

<https://1drv.ms/v/s!Avd4fnSa-w6DgWsUPJeo23zBe7pv>

Or

<https://photos.onedrive.com/share/830EFB9A747E78F7!235?cid=830EFB9A747E78F7&resId=830EFB9A747E78F7!235&authkey=!ABQ8l6jbfMF7um8&ithint=video>

---

Purushotham Palla