UNIVERSITY OF THE PUNJAB

Roll No. .....

Course Code: DI-322-L **Time Allowed: 2 Hours**
Course: Web Technologies Lab **Max. Marks: 100**

A **University Library Management** System requires an online **Book Borrowing Portal** where students can borrow books by **registering** themselves. Your task is to develop a basic version of this system with proper validation, backend logic, and database integration using **HTML**, **JavaScript**, **Java Servlets, and MySQL**.

Question # 1: Designing the Registration Form **[20]**

You are required to create an **HTML form** for student registration with the following fields:

1. **Full Name** (Text input)
2. **Email Address** (Text input)
3. **Student id** (Text input)
4. **Password** (Password input)
5. **Confirm Password** (Password input)
6. **Submit Button**

Question # 2: Implementing Frontend Validation with JavaScript **[20]**

Before allowing form submission, validate the inputs using **JavaScript** to enforce the following rules:

1. **Full Name:** Should contain only letters (e.g., "John Doe").
2. **Email Address:** Must follow the format "example@gmail.com".
3. **Student id:** Should be **6 characters** long.
4. **Password:** At least **8 characters**, with **one uppercase letter, one lowercase letter, and one digit**.
5. **Confirm Password:** Must match the password.

If validation fails, display appropriate **error messages below the respective fields**.

Question # 3: Create Database and Table **[20]**

Create a **MySQL database** named **Library** and a table **students** with the following Field:

1. **id** (INT AUTO_INCREMENT PRIMARY KEY)
2. **full_name** (VARCHAR(100) NOT NULL)
3. **student_id** (VARCHAR(100) NOT NULL)
4. **email** (VARCHAR(100) UNIQUE NOT NULL)
5. **password** (VARCHAR(255) NOT NULL)

Question # 4: Storing Data in Database and Handling Requests in Java Servlets **[20]**

Write a **Java Servlet (Register.java)** that:

1. Retrieves **form data** from the request.
2. **Connects to MySQL** and inserts user data into the students table.
3. **Check if the email already exists** in the database.
4. If the **user is successfully registered**, display **"Registration Successful!"** on the same page.
5. If registration **fails (e.g., email exists , student_id not exist or database error)**, display **"Registration Failed. Try Again!"** on the same page.

Viva. [20]