**SpringBoot\_Upgrading Registration Form In Spring Boot**

**Account Model**

package org.studyeasy.SpringBlog.models;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.OneToMany;

import javax.validation.constraints.Email;

import javax.validation.constraints.NotEmpty;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Entity

@Getter

@Setter

@NoArgsConstructor

public class Account {

    @Id

    @GeneratedValue(strategy = GenerationType.SEQUENCE)

    private Long id;

    @Email(message = "Invalid email")

    @NotEmpty(message = "Email missing")

    private String email;

    @NotEmpty(message = "Missing password")

    private String password;

    private String firstname;

    private String lastname;

    private String role;

    @OneToMany(mappedBy = "account")

    private List<Post> posts;

    @ManyToMany

    @JoinTable(

        name="account\_authority",

        joinColumns = {@JoinColumn(name="account\_id", referencedColumnName = "id")},

        inverseJoinColumns = {@JoinColumn(name = "authority\_id", referencedColumnName = "id")})

    private Set<Authority> authorities = new HashSet<>();

}

**Add below properties**

**1. Dropdown – Gender**

private String gender;

**2.Age – Number**

 private int age;

**3.Date Of Birth – LocalDate**

 @DateTimeFormat(pattern = "yyyy-MM-dd")

 private LocalDate date\_of\_birth;

**4.Photo**

 private String photo;

 @Value("${spring.mvc.static-path-pattern}")

 private String photo\_prefix;

**Updated code of Account.java -> Model**

package org.studyeasy.SpringBlog.models;

import java.time.LocalDate;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.OneToMany;

import javax.validation.constraints.Email;

import javax.validation.constraints.Max;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotEmpty;

import org.springframework.format.annotation.DateTimeFormat;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Entity

@Getter

@Setter

@NoArgsConstructor

public class Account {

    @Id

    @GeneratedValue(strategy = GenerationType.SEQUENCE)

    private Long id;

    @Email(message = "Invalid email")

    @NotEmpty(message = "Email missing")

    private String email;

    @NotEmpty(message = "Missing password")

    private String password;

    @NotEmpty(message = "Missing firstname")

    private String firstname;

    @NotEmpty(message = "Missing lastname")

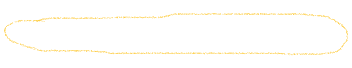
    private String lastname;

    private String gender;

    @Min(value=18)

    @Max(value=99)

    private int age;

    @DateTimeFormat(pattern = "yyyy-MM-dd")

    private LocalDate date\_of\_birth;

    private String photo;

    private String role;

    @OneToMany(mappedBy = "account")

    private List<Post> posts;

    @ManyToMany

    @JoinTable(

        name="account\_authority",

        joinColumns = {@JoinColumn(name="account\_id", referencedColumnName = "id")},

        inverseJoinColumns = {@JoinColumn(name = "authority\_id", referencedColumnName = "id")})

    private Set<Authority> authorities = new HashSet<>();

}

**Updated code of AccountService.java**

@Value("${spring.mvc.static-path-pattern}")

private String photo\_prefix;

 public Account save(Account account){

        account.setPassword(passwordEncoder.encode(account.getPassword()));

        if (account.getRole() == null){

            account.setRole(Roles.USER.getRole());

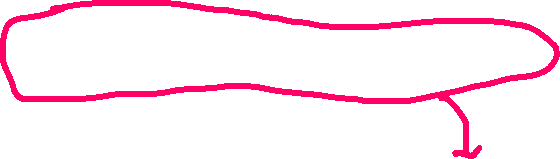
        }

        if (account.getPhoto() == null){

            String path = photo\_prefix.replace("\*\*", "images/person.png");

            account.setPhoto(path);

        }

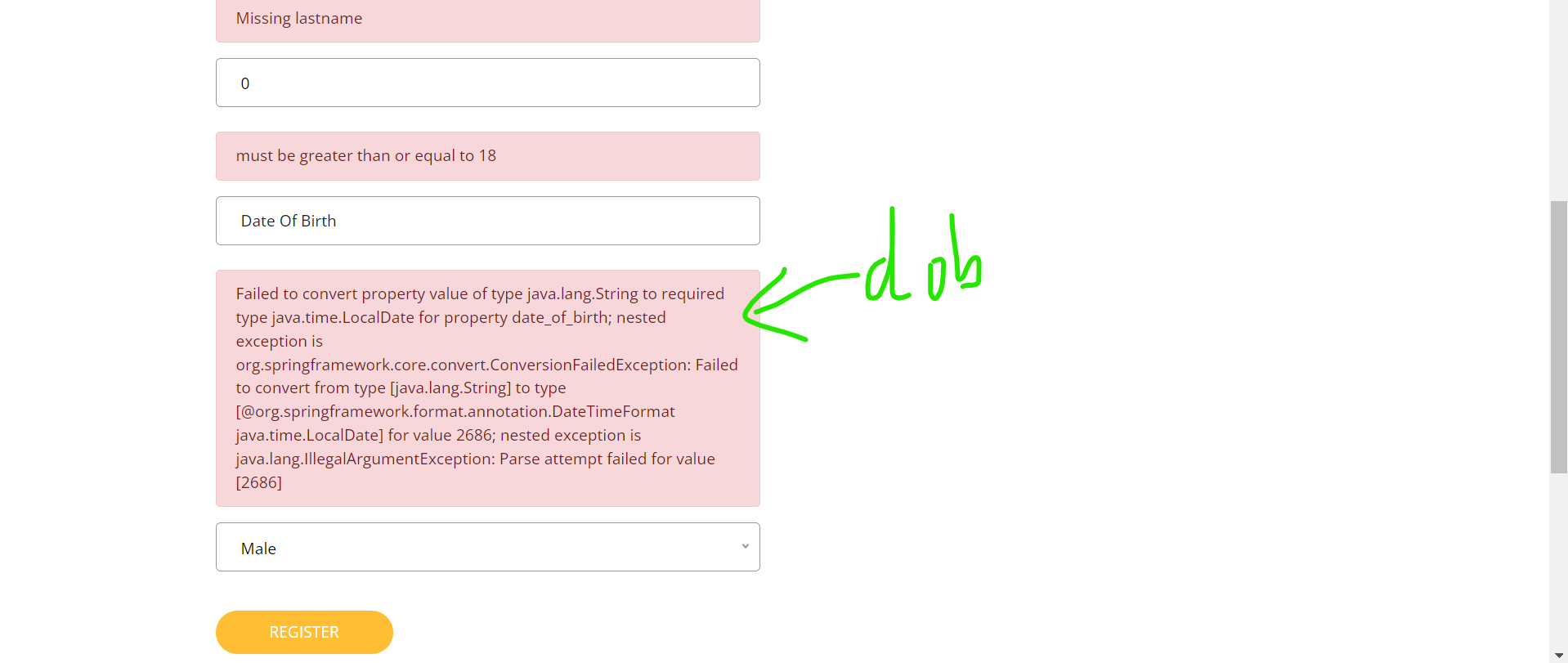


        return accountRepository.save(account);

    }

Image Condition

Date Problem



Change the type to date

 <div>



    <input id="date\_of\_birth" type="date" th:field="\*{date\_of\_birth}" class="form-control" placeholder="Date Of Birth" />

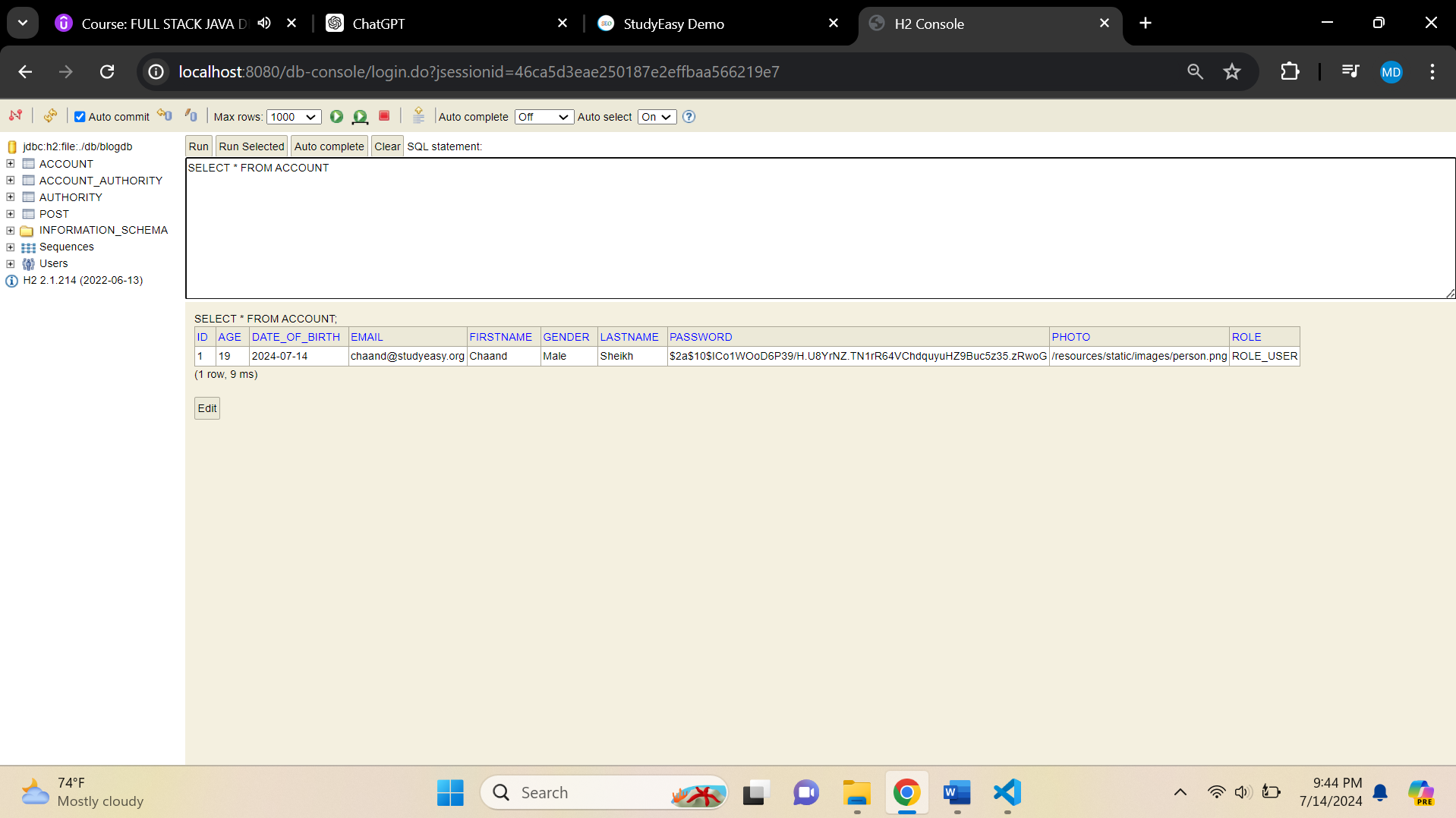


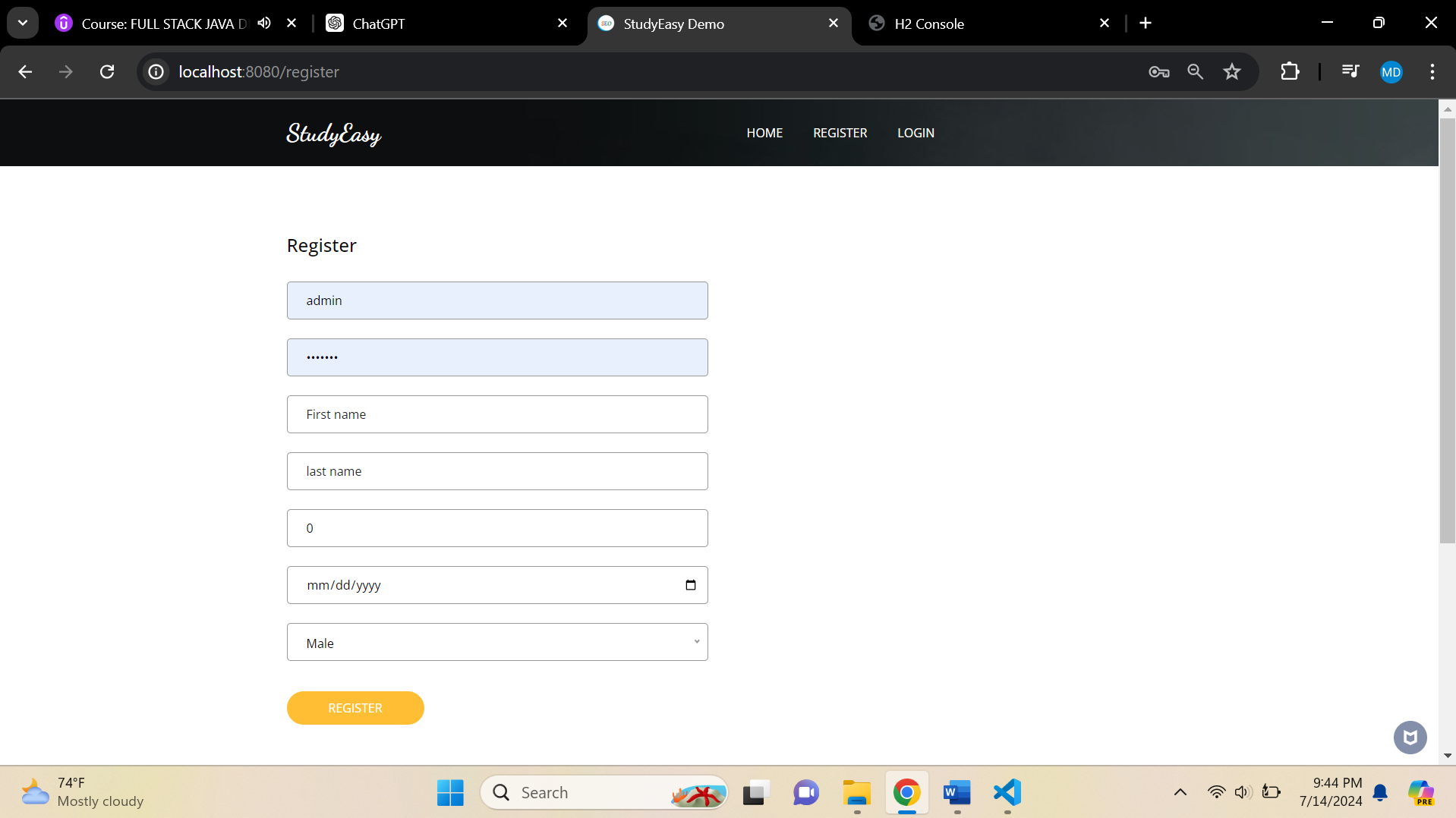
    <p th:if="${#fields.hasErrors('date\_of\_birth')}" th:errors="\*{date\_of\_birth}" class="alert alert-danger" role="alert">date of birth Error</p>

</div>

Client-Side Validations will help to speed up application – required attribute

Output





**Updated code of register.html**

<!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

  <head th:replace="/fragments/head :: head">

<body class="sub\_page">

  <div class="hero\_area">

    <div class="bg-box">

      <img th:src="@{/resources/static/images/nav-bg.jpg}" alt="">

    </div>

    <header th:replace="/fragments/header :: header">

  </div>

 <!-- book section -->

 <section class="book\_section layout\_padding">

    <div class="container">

      <div class="heading\_container">

        <h4>

         Register

        </h4>

      </div>

      <div class="row">

        <div class="col-md-6">

          <div class="form\_container">

            <form action="#"

                th:action="@{/register}"

                th:object="${account}"

                method="post">

              <div>

                <input id="email" name="email" type="text" th:field="\*{email}" class="form-control" placeholder="Email" required/>

                <p th:if="${#fields.hasErrors('email')}" th:errors="\*{email}" class="alert alert-danger" role="alert">Email Error</p>

              </div>

              <div>

                <input id="password" type="password" th:field="\*{password}" class="form-control" placeholder="Password" required/>

                <p th:if="${#fields.hasErrors('password')}" th:errors="\*{password}" class="alert alert-danger" role="alert">password Error</p>

              </div>

              <div>

                <input id="firstname" type="text" th:field="\*{firstname}" class="form-control" placeholder="First name" required/>

                <p th:if="${#fields.hasErrors('firstname')}" th:errors="\*{firstname}" class="alert alert-danger" role="alert">firstname Error</p>

              </div>

              <div>

                <input id="lastname" type="text" th:field="\*{lastname}" class="form-control" placeholder="last name" required />

                <p th:if="${#fields.hasErrors('lastname')}" th:errors="\*{lastname}" class="alert alert-danger" role="alert">lastname Error</p>

              </div>

              <div>

                <input id="age" type="number" th:field="\*{age}" class="form-control" placeholder="age" min="18" max="99" required/>

                <p th:if="${#fields.hasErrors('age')}" th:errors="\*{age}" class="alert alert-danger" role="alert">age Error</p>

              </div>

              <div>

                <input id="date\_of\_birth" type="date" th:field="\*{date\_of\_birth}" class="form-control" placeholder="Date Of Birth" required/>

                <p th:if="${#fields.hasErrors('date\_of\_birth')}" th:errors="\*{date\_of\_birth}" class="alert alert-danger" role="alert">date of birth Error</p>

              </div>

              <div>

                <select id="gender" name="gender" th:field="\*{gender}" class="form-control">

                  <option value="Male">Male</option>

                  <option value="Female">Female</option>

                  <option value="Others">Others</option>

                </select>

                <p th:if="${#fields.hasErrors('gender')}" th:errors="\*{gender}" class="alert alert-danger" role="alert">gender Error</p>

              </div>

              <div class="btn\_box">

                <button type="submit">

                  Register

                </button>

              </div>

            </form>

          </div>

        </div>

    </div>

  </section>

  <!-- end book section -->

  <footer th:replace="/fragments/footer :: footer">

</body>

</html>