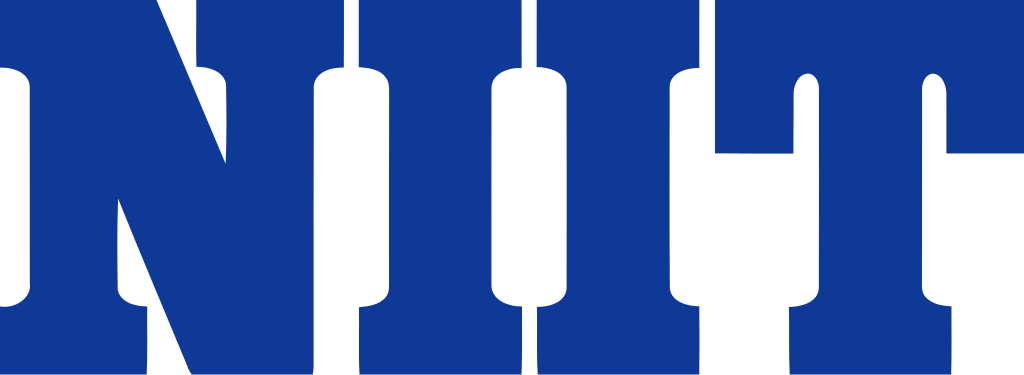
PROJECT ON

**Online Collaboration**



**Developed By: - Sandip Mondal**

**Online Collaboration**

Batch Code : S210192

Start Date : 23rd February 2021

End Date : 14th April 2021

Name Of The Coordinator : Mrs. Lopamudra Bera

Name Of The Developer : Sandip Mondal

Date Of Submission : 14th April 2021

**Online Collaboration**

CERTIFICATE

This is to certify that this report, titled Online Collaboration embodies the original work done by Sandip Mondal, in partial fulfilment of their course requirement at NIIT.

CO-ORDINATOR:

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Mrs. Lopamudra Bera who gave me the golden opportunity to do this wonderful project on the topic Online Collaboration Application, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Configuration

Hardware

Processor : Intel i3 or higher.

RAM : 2GB (minimum)

Speed : 1.5GHz

Secondary

Storage : 10GB

Software :

Java Version : JDK 1.8

Database

Management: MySQL

IDE : Eclipse

OS : Windows 10.

Steps with codes:

* **Database in MySQL:**

1. Open MySQL Workbench
2. Creating a schema “collaborate”
3. Activating schema

use collaborate;

1. Creating table “User”

create table User(

UserId int not null auto\_increment,

FirstName varchar(30),

LastName varchar(30),

UserName varchar(20),

Password varchar(20),

email varchar(40),

Role varchar(5),

Status varchar(10),

IsOnline boolean,

Enabled boolean,

primary key(UserId)

);

1. Insert one record with Admin role into User table:

insert into User(FirstName, LastName, UserName, Password, email, Role, Status, IsOnline, Enabled) values ('Sandip', 'Mondal', 'sandip1', 'sandip1234', 'sandipmondal990355@gmail.com','Admin', 'Active', true, true);

1. Creating another table “Blog”

create table Blog(

BlogId int not null auto\_increment,

BlogTitle varchar(30),

BlogContent varchar(200),

BlogPosted Date,

status varchar(10),

NoOfLikes int,

NoOfViews int,

NoOfComments int,

UserId int,

Username varchar(20),

primary key(BlogId)

);

1. Creating another table “BlogComments”

create table BlogComments(

BlogCommentId int not null auto\_increment,

UserId int,

Username varchar(20),

UserProfileId varchar(20),

Title varchar(30),

NonOfLikes int,

BlogComment varchar(50),

CurrentDate Date,

BlogId int,

primary key(BlogCommentId)

);

* **Backend Code in Eclipse IDE:**

1. Open any chrome browser
2. Click on the URL tag and paste the link <https://start.spring.io/> and visit the page
3. Creating a SpringBoot project named “OnlineCollaborate” (You can suggest any feasible project name according to project specification) with web, Spring Data JPA, SpringBoot Dev Tools and MySQL Server Driver packages. Extract that project.
4. Importing the project in Eclipse.
5. Creating a configuration class “HibernateConfig.java” inside com.coll.OnlineCollaborate.config package and specifying the required configuration in it.

package com.coll.OnlineCollaboration.config;

import java.util.Properties;

import javax.sql.DataSource;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.boot.autoconfigure.orm.jpa.HibernateJpaAutoConfiguration;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.ComponentScans;

import org.springframework.context.annotation.Configuration;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.orm.hibernate5.HibernateTransactionManager;

import org.springframework.orm.hibernate5.LocalSessionFactoryBean;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import org.springframework.web.servlet.ViewResolver;

import org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration

@ComponentScans(value= {@ComponentScan("com.coll.OnlineCollaboration"),

@ComponentScan("model"),

@ComponentScan("controller"),

@ComponentScan("dao"),

@ComponentScan("service")})

@EnableAutoConfiguration(exclude = { HibernateJpaAutoConfiguration.class})

@EnableTransactionManagement

public class HibernateConfig {

public static final String DATABASE\_URL="jdbc:mysql://localhost:3306/collaborate";

public static final String DATABASE\_DRIVER="com.mysql.cj.jdbc.Driver";

public static final String DATABASE\_DIALECT="org.hibernate.dialect.MySQLDialect";

public static final String DATABASE\_USERNAME="root";

public static final String DATABASE\_PASSWORD="niit@1234";

@Bean(name="dataSource")

public DataSource getDataSource() {

DriverManagerDataSource dataSource=new DriverManagerDataSource();

dataSource.setDriverClassName(DATABASE\_DRIVER);

dataSource.setUrl(DATABASE\_URL);

dataSource.setUsername(DATABASE\_USERNAME);

dataSource.setPassword(DATABASE\_PASSWORD);

return dataSource;

}

@Bean

public LocalSessionFactoryBean getSessionFactory() {

LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();

sessionFactory.setDataSource(getDataSource());

sessionFactory.setPackagesToScan("com.coll.OnlineCollaboration");

Properties hibernateProperties = new Properties();

hibernateProperties.put("hibernate.dialect", DATABASE\_DIALECT);

hibernateProperties.put("hibernate.show\_sql", "true");

hibernateProperties.put("hibernate.hbm2ddl.auto", "update");

sessionFactory.setHibernateProperties(hibernateProperties);

return sessionFactory;

}

@Bean

public HibernateTransactionManager getTransactionManager() {

HibernateTransactionManager txm=new HibernateTransactionManager();

txm.setSessionFactory(getSessionFactory().getObject());

return txm;

}

@Bean

public ViewResolver jspViewResolver() {

InternalResourceViewResolver viewResolver=new InternalResourceViewResolver();

viewResolver.setPrefix("/views/");

viewResolver.setSuffix(".jsp");

return viewResolver;

}

}

1. Creating an Entity/POJO (Plain Old Java Object) class inside com.coll.OnlineCollaboration.model package. Name of the class is “User”

package com.coll.OnlineCollaboration.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import org.springframework.stereotype.Component;

@Component

@Entity

public class User extends DomainResponse implements Serializable {

private static final long *serialVersionUID*=1L;

@Id

@GeneratedValue(strategy=GenerationType.*IDENTITY*)

private int userId;

private String firstName;

private String lastName;

private String username;

private String password;

private String email;

private String role;

private String status;

private boolean isOnline;

private boolean enabled;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public boolean isOnline() {

return isOnline;

}

public void setOnline(boolean isOnline) {

this.isOnline = isOnline;

}

public boolean isEnabled() {

return enabled;

}

public void setEnabled(boolean enabled) {

this.enabled = enabled;

}

public static long getSerialversionuid() {

return *serialVersionUID*;

}

}

1. Creating another class inside the above said package named “Domain Response”.

package com.coll.OnlineCollaboration.model;

public class DomainResponse {

int responseCode;

String responseMessage;

public DomainResponse() {

super();

// TODO Auto-generated constructor stub

}

public DomainResponse(int responseCode, String responseMessage) {

super();

this.responseCode = responseCode;

this.responseMessage = responseMessage;

}

public int getResponseCode() {

return responseCode;

}

public void setResponseCode(int responseCode) {

this.responseCode = responseCode;

}

public String getResponseMessage() {

return responseMessage;

}

public void setResponseMessage(String responseMessage) {

this.responseMessage = responseMessage;

}

}

1. Create another class inside the above said package named “Blog”.

package com.coll.OnlineCollaboration.model;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import org.springframework.stereotype.Component;

import com.fasterxml.jackson.annotation.JsonManagedReference;

@Component

@Entity

public class Blog extends DomainResponse implements Serializable{

private static final long serialVersionUID=1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

int blogId;

String blogTitle, blogContent;

LocalDate blogPosted;

String status;

int noOfLikes, noOfComponents, noOfViews;

int userId;

String username;

@OneToMany(mappedBy="blog",fetch=FetchType.EAGER,cascade=CascadeType.ALL)

@JsonManagedReference

List<BlogComments>blogComments;

public int getBlogId() {

return blogId;

}

public void setBlogId(int blogId) {

this.blogId = blogId;

}

public String getBlogTitle() {

return blogTitle;

}

public void setBlogTitle(String blogTitle) {

this.blogTitle = blogTitle;

}

public String getBlogContent() {

return blogContent;

}

public void setBlogContent(String blogContent) {

this.blogContent = blogContent;

}

public LocalDate getBlogPosted() {

return blogPosted;

}

public void setBlogPosted(LocalDate blogPosted) {

this.blogPosted = blogPosted;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public int getNoOfLikes() {

return noOfLikes;

}

public void setNoOfLikes(int noOfLikes) {

this.noOfLikes = noOfLikes;

}

public int getNoOfComponents() {

return noOfComponents;

}

public void setNoOfComponents(int noOfComponents) {

this.noOfComponents = noOfComponents;

}

public int getNoOfViews() {

return noOfViews;

}

public void setNoOfViews(int noOfViews) {

this.noOfViews = noOfViews;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public List<BlogComments> getBlogComments() {

return blogComments;

}

public void setBlogComments(List<BlogComments> blogComments) {

this.blogComments = blogComments;

}

public static long getSerialversionuid() {

return serialVersionUID;

}

}

1. Create another class inside the above said package named “BlogComments”.

package com.coll.OnlineCollaboration.model;

import java.io.Serializable;

import java.time.LocalDate;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import com.fasterxml.jackson.annotation.JsonBackReference;

public class BlogComments implements Serializable{

private static final long serialVersionUID=1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

int blogCommentedId;

int userId;

String username;

String userProfileId;

String title;

int noOfLikes;

String blogComment;

LocalDate currentDate;

@ManyToOne

@JoinColumn(name="BlogId")

@JsonBackReference

Blog blog;

public int getBlogCommentedId() {

return blogCommentedId;

}

public void setBlogCommentedId(int blogCommentedId) {

this.blogCommentedId = blogCommentedId;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getUserProfileId() {

return userProfileId;

}

public void setUserProfileId(String userProfileId) {

this.userProfileId = userProfileId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public int getNoOfLikes() {

return noOfLikes;

}

public void setNoOfLikes(int noOfLikes) {

this.noOfLikes = noOfLikes;

}

public String getBlogComment() {

return blogComment;

}

public void setBlogComment(String blogComment) {

this.blogComment = blogComment;

}

public LocalDate getCurrentDate() {

return currentDate;

}

public void setCurrentDate(LocalDate currentDate) {

this.currentDate = currentDate;

}

public Blog getBlog() {

return blog;

}

public void setBlog(Blog blog) {

this.blog = blog;

}

public static long getSerialversionuid() {

return serialVersionUID;

}

}

1. Create the DAO interfaces inside com.coll.OnlineCollaborate.dao package. Create an interface named IUserDao.java

package com.coll.OnlineCollaboration.dao;

import java.util.List;

import com.coll.OnlineCollaboration.model.User;

public interface IUserDao {

List<User> userListbyStatus(String status);

List<User> getAllUsers();

User getUserById(int userId);

User getUserByUsername(String username);

User validateUser(User user);

boolean addUser(User user);

boolean updateUser(User user);

boolean deleteUser(int userId);

boolean deactiveUser(int userId);

boolean updateUserProfile(String file, Integer userId);

}

1. Create the DAO interfaces inside com.coll.OnlineCollaborate.dao package. Create an interface named IBlogDao.java

package com.coll.OnlineCollaboration.dao;

import java.util.List;

import com.coll.OnlineCollaboration.model.Blog;

public interface IBlogDao {

List<Blog> getAllBlogs();

List<Blog> getBlogsByStatus(String status);

List<Blog> getUsersBlogs(int id);

List<Blog> mainList();

Blog getBlogById(int blogId);

boolean addBlog(Blog blog);

boolean updateBlog(Blog blog);

boolean deleteBlog(Blog blog);

}

1. Create the DAO interfaces inside com.coll.OnlineCollaborate.dao package. Create an interface named IBlogCommentsDao.java

package com.coll.OnlineCollaboration.dao;

import java.util.List;

import com.coll.OnlineCollaboration.model.BlogComments;

public interface IBlogCommentsDao {

List<BlogComments> getAllBlogComments();

BlogComments getBlogCommentsById(int blogComemntId);

boolean addBlogComments(BlogComments blogComments);

boolean updateBlogComments(BlogComments blogComments);

boolean deleteBlogComments(BlogComments blogComments);

}

1. Create the DAO interfaces inside com.coll.OnlineCollaborate.dao package. Create an interface named IBlogDao.java

package com.coll.OnlineCollaboration.dao;

import java.util.List;

import com.coll.OnlineCollaboration.model.Blog;

public interface IBlogDao {

List<Blog> getAllBlogs();

List<Blog> getBlogsByStatus(String status);

List<Blog> getUsersBlogs(int userId);

List<Blog> mainList();

Blog getBlogById(int blogId);

boolean addBlog(Blog blog);

boolean updateBlog(Blog blog);

boolean deleteBlog(Blog blog);

}

1. Create the DAO interfaces inside com.coll.OnlineCollaborate.dao package. Create an interface named IUserDao.java

package com.coll.OnlineCollaboration.dao;

import java.util.List;

import com.coll.OnlineCollaboration.model.User;

public interface IUserDao {

List<User> userListbyStatus(String status);

List<User> getAllUsers();

User getUserById(int userId);

User getUserByUsername(String username);

User validateUser(User user);

boolean addUser(User user);

boolean updateUser(User user);

boolean deleteUser(int userId);

boolean deactivateUser(int userId);

List<User> getAllInactivateUsers();

boolean activeUser(int userId);

boolean updateUserProfile(String file, Integer userId);

}

1. Create the class named BlogCommentsDaoImpl.java inside com.coll.OnlineCollaboration.daoImpl package.

package com.coll.OnlineCollaboration.daoImpl;

import java.util.List;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IBlogCommentsDao;

import com.coll.OnlineCollaboration.model.BlogComments;

@Repository("blogCommentsDao")

@Transactional

public class BlogCommentsDaoImpl implements IBlogCommentsDao{

@Autowired

SessionFactory sessionFactory;

@Override

public List<BlogComments> getAllBlogComments() {

return sessionFactory.getCurrentSession().createQuery("from BlogComments", BlogComments.class).getResultList();

}

@Override

public BlogComments getBlogCommentsById(int blogCommentId) {

return sessionFactory.getCurrentSession().get(BlogComments.class, Integer.valueOf(blogCommentId));

}

@Override

public boolean addBlogComments(BlogComments blogComments) {

try

{

sessionFactory.getCurrentSession().save(blogComments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateBlogComments(BlogComments blogComments) {

try

{

sessionFactory.getCurrentSession().update(blogComments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteBlogComments(BlogComments blogComments) {

try

{

sessionFactory.getCurrentSession().delete(blogComments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

}

1. Create the class named BlogDaoImpl.java inside com.coll.OnlineCollaboration.daoImpl package.

package com.coll.OnlineCollaboration.daoImpl;

import java.util.List;

import org.hibernate.query.Query;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IBlogDao;

import com.coll.OnlineCollaboration.model.Blog;

@Repository("blogDao")

@Transactional

public class BlogDaoImpl implements IBlogDao{

@Autowired

SessionFactory sessionFactory;

@Override

public List<Blog> getAllBlogs() {

return sessionFactory.getCurrentSession().createQuery("from Blog", Blog.class).getResultList();

}

@Override

public List<Blog> getBlogsByStatus(String status) {

String q="from Blog where status='"+status+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

@Override

public List<Blog> getUsersBlogs(int userId) {

String q="from Blog where UserId='"+userId+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

@Override

public List<Blog> mainList() {

return sessionFactory.getCurrentSession().createQuery("from Blog", Blog.class).getResultList();

}

@Override

public Blog getBlogById(int blogId) {

return sessionFactory.getCurrentSession().get(Blog.class,Integer.valueOf(blogId));

}

@Override

public boolean addBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().save(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().update(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().delete(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

}

1. Create the class named UserDaoImpl.java inside com.coll.OnlineCollaboration.daoImpl package.

package com.coll.OnlineCollaboration.daoImpl;

import java.util.List;

import org.hibernate.query.Query;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IUserDao;

import com.coll.OnlineCollaboration.model.User;

@Repository("userDao")

@Transactional

public class UserDaoImpl implements IUserDao{

@Autowired

SessionFactory sessionFactory;

@Override

public List<User> userListbyStatus(String status) {

String q="from User where status='"+status+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

@Override

public List<User> getAllUsers() {

return sessionFactory.getCurrentSession().createQuery("from User",User.class).getResultList();

}

@Override

public User getUserById(int userId) {

return sessionFactory.getCurrentSession().get(User.class, Integer.valueOf(userId));

}

@Override

public User getUserByUsername(String username) {

String query="from User where username=:username";

return sessionFactory.getCurrentSession().createQuery(query, User.class).setParameter("username", username).getSingleResult();

}

@Override

public User validateUser(User user) {

String username=user.getUsername();

String password=user.getPassword();

String q="From User where username='"+username+"' and password='"+password+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

try

{

user=(User)query.getSingleResult();

user.setIsOnline(true);

return user;

}

catch(Exception e)

{

e.printStackTrace();

return null;

}

}

@Override

public boolean addUser(User user) {

try

{

sessionFactory.getCurrentSession().save(user);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateUser(User user) {

try

{

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteUser(int userId) {

try

{

sessionFactory.getCurrentSession().delete(getUserById(userId));

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deactivateUser(int userId) {

try

{

User user=getUserById(userId);

user.setEnabled(false);

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public List<User> getAllInactivateUsers() {

return sessionFactory.getCurrentSession().createQuery("from User where enabled=false",User.class).getResultList();

}

@Override

public boolean activeUser(int userId) {

try

{

User user=getUserById(userId);

user.setEnabled(true);

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateUserProfile(String file, Integer userId) {

String q="update User set profile=:fileName where userId=:id";

Query query=sessionFactory.getCurrentSession().createQuery(q);

query.setParameter("id", (Integer)userId);

query.setParameter("fileName", file);

try

{

query.executeUpdate();

return true;

}

catch(Exception e)

{

e.printStackTrace();

return false;

}

}

}

1. Create model class inside the com.coll.OnlineCollaboration.model package. Create Blog.java

package com.coll.OnlineCollaboration.model;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import org.springframework.stereotype.Component;

import com.fasterxml.jackson.annotation.JsonManagedReference;

@Component

@Entity

public class Blog extends DomainResponse implements Serializable{

private static final long serialVersionUID=1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

int blogId;

String blogTitle, blogContent;

LocalDate blogPosted;

String status;

int noOfLikes, noOfViews, noOfComments;

int userId;

String username;

@OneToMany(mappedBy="blog", fetch=FetchType.EAGER, cascade=CascadeType.ALL)

@JsonManagedReference

List<BlogComments> blogComments;

public int getBlogId() {

return blogId;

}

public void setBlogId(int blogId) {

this.blogId = blogId;

}

public String getBlogTitle() {

return blogTitle;

}

public void setBlogTitle(String blogTitle) {

this.blogTitle = blogTitle;

}

public String getBlogContent() {

return blogContent;

}

public void setBlogContent(String blogContent) {

this.blogContent = blogContent;

}

public LocalDate getBlogPosted() {

return blogPosted;

}

public void setBlogPosted(LocalDate blogPosted) {

this.blogPosted = blogPosted;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public int getNoOfLikes() {

return noOfLikes;

}

public void setNoOfLikes(int noOfLikes) {

this.noOfLikes = noOfLikes;

}

public int getNoOfComments() {

return noOfComments;

}

public void setNoOfComments(int noOfComments) {

this.noOfComments = noOfComments;

}

public int getNoOfViews() {

return noOfViews;

}

public void setNoOfViews(int noOfViews) {

this.noOfViews = noOfViews;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public List<BlogComments> getBlogComments() {

return blogComments;

}

public void setBlogComments(List<BlogComments> blogComments) {

this.blogComments = blogComments;

}

public static long getSerialversionuid() {

return serialVersionUID;

}

}

1. Create model class inside the com.coll.OnlineCollaboration.model package. Create BlogComments.java

**package** com.coll.OnlineCollaboration.model;

**import** java.io.Serializable;

**import** java.time.LocalDate;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.ManyToOne;

**import** org.springframework.stereotype.Component;

**import** com.fasterxml.jackson.annotation.JsonBackReference;

@Component

@Entity

**public** **class** BlogComments **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID***=1L;

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**int** blogCommentId;

**int** userId;

String username;

String userProfileId;

String title;

**int** noOfLikes;

String blogComment;

LocalDate currentDate;

@ManyToOne

@JoinColumn(name="BlogId")

@JsonBackReference

Blog blog;

**public** **int** getBlogCommentId() {

**return** blogCommentId;

}

**public** **void** setBlogCommentId(**int** blogCommentId) {

**this**.blogCommentId = blogCommentId;

}

**public** **int** getUserId() {

**return** userId;

}

**public** **void** setUserId(**int** userId) {

**this**.userId = userId;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getUserProfileId() {

**return** userProfileId;

}

**public** **void** setUserProfileId(String userProfileId) {

**this**.userProfileId = userProfileId;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** **int** getNoOfLikes() {

**return** noOfLikes;

}

**public** **void** setNoOfLikes(**int** noOfLikes) {

**this**.noOfLikes = noOfLikes;

}

**public** String getBlogComment() {

**return** blogComment;

}

**public** **void** setBlogComment(String blogComment) {

**this**.blogComment = blogComment;

}

**public** LocalDate getCurrentDate() {

**return** currentDate;

}

**public** **void** setCurrentDate(LocalDate currentDate) {

**this**.currentDate = currentDate;

}

**public** Blog getBlog() {

**return** blog;

}

**public** **void** setBlog(Blog blog) {

**this**.blog = blog;

}

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

}

1. Create model class inside the com.coll.OnlineCollaboration.model package. Create DomainResponse.java

**package** com.coll.OnlineCollaboration.model;

**public** **class** DomainResponse {

**int** responseCode;

String responseMessage;

**public** DomainResponse() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** DomainResponse(**int** responseCode, String responseMessage) {

**super**();

**this**.responseCode = responseCode;

**this**.responseMessage = responseMessage;

}

**public** **int** getResponseCode() {

**return** responseCode;

}

**public** **void** setResponseCode(**int** responseCode) {

**this**.responseCode = responseCode;

}

**public** String getResponseMessage() {

**return** responseMessage;

}

**public** **void** setResponseMessage(String responseMessage) {

**this**.responseMessage = responseMessage;

}

}

1. Create model class inside the com.coll.OnlineCollaboration.model package. Create User.java

package com.coll.OnlineCollaboration.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import org.springframework.stereotype.Component;

@Component

@Entity

public class User extends DomainResponse implements Serializable{

private static final long serialVersionUID=1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

private int userId;

private String firstName;

private String lastName;

private String username;

private String password;

private String email;

private String role;

private String status;

private boolean isOnline;

private boolean enabled;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public boolean getIsOnline() {

return isOnline;

}

public void setIsOnline(boolean isOnline) {

this.isOnline = isOnline;

}

public boolean getEnabled() {

return enabled;

}

public void setEnabled(boolean enabled) {

this.enabled = enabled;

}

}

1. Create interface in com.coll.OnlineCollaboration.service packge.

Create IBlogCommentService.java

package com.coll.OnlineCollaboration.service;

import java.util.List;

import com.coll.OnlineCollaboration.model.BlogComments;

public interface IBlogCommentsService {

List<BlogComments> getAllBlogComments();

BlogComments getBlogCommentsById(int blogCommentId);

boolean addBlogComments(BlogComments blogComments);

boolean updateBlogComments(BlogComments blogComments);

boolean deleteBlogComments(BlogComments blogComments);

}

1. Create interface in com.coll.OnlineCollaboration.service packge.

Create IBlogService.java

package com.coll.OnlineCollaboration.service;

import java.util.List;

import com.coll.OnlineCollaboration.model.Blog;

public interface IBlogService {

List<Blog> getAllBlogs();

List<Blog> getBlogsByStatus(String status);

List<Blog> getUsersBlogs(int id);

List<Blog> mainList();

Blog getBlogById(int blogId);

boolean addBlog(Blog blog);

boolean updateBlog(Blog blog);

boolean deleteBlog(Blog blog);

}

1. Create interface in com.coll.OnlineCollaboration.service packge.

Create IUserService.java

package com.coll.OnlineCollaboration.service;

import java.util.List;

import com.coll.OnlineCollaboration.model.User;

public interface IUserService {

List<User> userListbyStatus(String status);

List<User> getAllUsers();

User getUserById(int userId);

User getUserByUsername(String username);

User validateUser(User user);

boolean addUser(User user);

boolean updateUser(User user);

boolean deleteUser(int userId);

boolean deactiveUser(int userId);

List<User> getAllInactivateUsers();

boolean activeUser(int userId);

boolean updateUserProfile(String file, Integer userId);

}

1. Create BlogCommentsServiceImpl.java inside the com.coll.OnlineCollaboration.serviceImpl package

package com.coll.OnlineCollaboration.serviceImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IBlogCommentsDao;

import com.coll.OnlineCollaboration.model.BlogComments;

import com.coll.OnlineCollaboration.service.IBlogCommentsService;

@Service

@Transactional

public class BlogCommentsServiceImpl implements IBlogCommentsService{

@Autowired

IBlogCommentsService blogCommentsService;

@Override

public List<BlogComments> getAllBlogComments() {

return blogCommentsService.getAllBlogComments();

}

@Override

public BlogComments getBlogCommentsById(int blogCommentId) {

return blogCommentsService.getBlogCommentsById(blogCommentId);

}

@Override

public boolean addBlogComments(BlogComments blogComments) {

return blogCommentsService.addBlogComments(blogComments);

}

@Override

public boolean updateBlogComments(BlogComments blogComments) {

return blogCommentsService.updateBlogComments(blogComments);

}

@Override

public boolean deleteBlogComments(BlogComments blogComments) {

return blogCommentsService.deleteBlogComments(blogComments);

}

}

1. Create BlogServiceImpl.java inside the com.coll.OnlineCollaboration.serviceImpl package

package com.coll.OnlineCollaboration.serviceImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IBlogDao;

import com.coll.OnlineCollaboration.model.Blog;

import com.coll.OnlineCollaboration.service.IBlogService;

@Service

@Transactional

public class BlogServiceImpl implements IBlogService{

@Autowired

IBlogDao blogDao;

@Override

public List<Blog> getAllBlogs() {

return blogDao.getAllBlogs();

}

@Override

public List<Blog> getBlogsByStatus(String status) {

return blogDao.getBlogsByStatus(status);

}

@Override

public List<Blog> getUsersBlogs(int id) {

return blogDao.getUsersBlogs(id);

}

@Override

public List<Blog> mainList() {

return blogDao.mainList();

}

@Override

public Blog getBlogById(int blogId) {

return blogDao.getBlogById(blogId);

}

@Override

public boolean addBlog(Blog blog) {

return blogDao.addBlog(blog);

}

@Override

public boolean updateBlog(Blog blog) {

return blogDao.updateBlog(blog);

}

@Override

public boolean deleteBlog(Blog blog) {

return blogDao.deleteBlog(blog);

}

}

1. Create UserServiceImpl.java inside the com.coll.OnlineCollaboration.serviceImpl package

package com.coll.OnlineCollaboration.serviceImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaboration.dao.IUserDao;

import com.coll.OnlineCollaboration.model.User;

import com.coll.OnlineCollaboration.service.IUserService;

@Service

@Transactional

public class UserServiceImpl implements IUserService{

@Autowired

IUserDao userDao;

@Override

public List<User> userListbyStatus(String status) {

return userDao.userListbyStatus(status);

}

@Override

public List<User> getAllUsers() {

return userDao.getAllUsers();

}

@Override

public User getUserById(int userId) {

return userDao.getUserById(userId);

}

@Override

public User getUserByUsername(String username) {

return userDao.getUserByUsername(username);

}

@Override

public User validateUser(User user) {

return userDao.validateUser(user);

}

@Override

public boolean addUser(User user) {

return userDao.addUser(user);

}

@Override

public boolean updateUser(User user) {

return userDao.updateUser(user);

}

@Override

public boolean deleteUser(int userId) {

return userDao.deleteUser(userId);

}

@Override

public boolean deactiveUser(int userId) {

return userDao.deactivateUser(userId);

}

@Override

public boolean updateUserProfile(String file, Integer userId) {

return userDao.updateUserProfile(file, userId);

}

@Override

public List<User> getAllInactivateUsers() {

return userDao.getAllInactivateUsers();

}

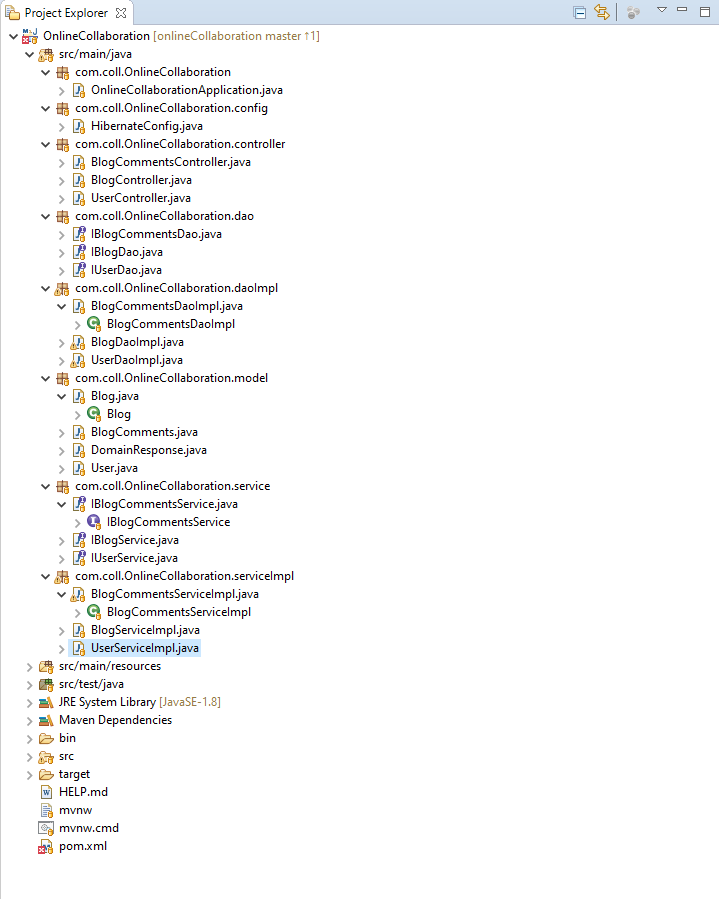
@Override

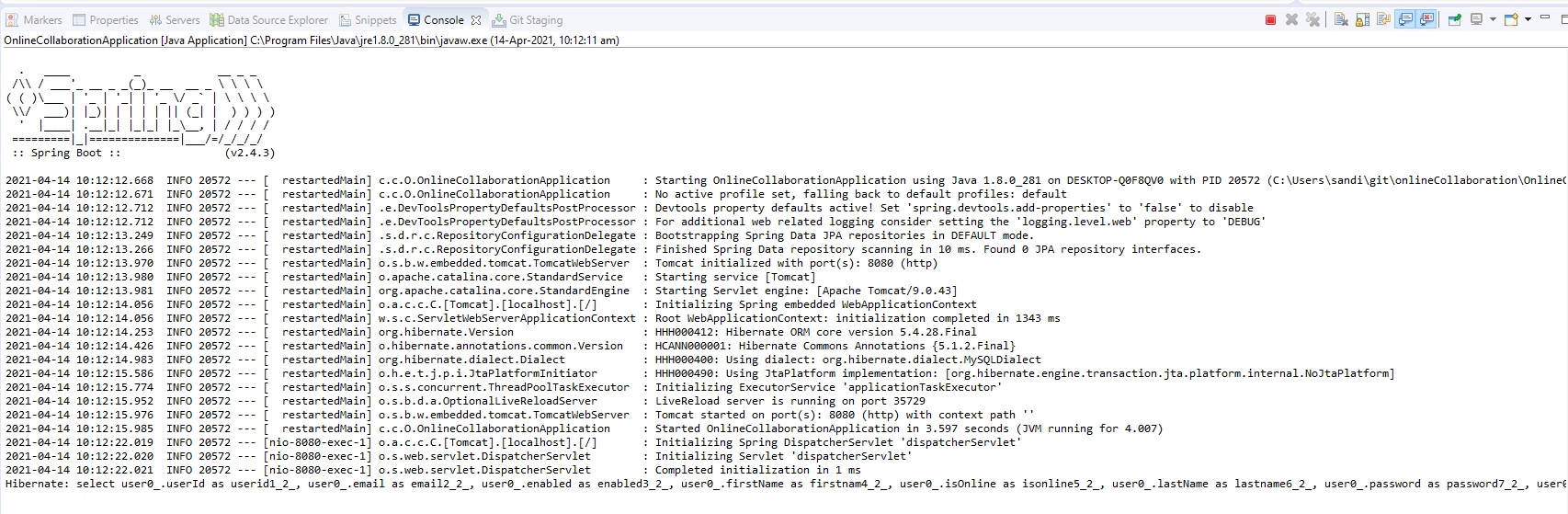
public boolean activeUser(int userId) {

return userDao.activeUser(userId);

}

}





**Frontend code**

Create angular project and implement the given frontend code:

1. app.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import {DataTablesModule} from 'angular-datatables';

import {FormsModule, ReactiveFormsModule} from '@angular/forms';

import {HttpClientModule} from '@angular/common/http';

import { RegisterUserComponent } from './components/register-user/register-user.component';

import { UserListComponent } from './components/user-list/user-list.component';

import { LoginUserComponent } from './components/login-user/login-user.component';

import { ActivateUserComponent } from './components/activate-user/activate-user.component';

import { AddBlogComponent } from './components/add-blog/add-blog.component';

import { ApproveBlogComponent } from './components/approve-blog/approve-blog.component';

import { ViewBlogComponent } from './components/view-blog/view-blog.component';

import {UserService} from './services/user.service';

import {BlogService} from './services/blog.service';

import { AdminHomeComponent } from './components/admin-home/admin-home.component';

@NgModule({

  declarations: [

    AppComponent,

    RegisterUserComponent,

    UserListComponent,

    LoginUserComponent,

    ActivateUserComponent,

    AddBlogComponent,

    ApproveBlogComponent,

    ViewBlogComponent,

    AdminHomeComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule,

    DataTablesModule,

    HttpClientModule,

    ReactiveFormsModule

  ],

  providers: [UserService, BlogService],

  bootstrap: [AppComponent]

})

export class AppModule { }

1. app.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.scss']

})

export class AppComponent {

  title = 'OnlineCollaborationAngular';

}

1. app.component.spec.ts

import { TestBed } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

import { AppComponent } from './app.component';

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'OnlineCollaborationAngular'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('OnlineCollaborationAngular');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement;

    expect(compiled.querySelector('.content span').textContent).toContain('OnlineCollaborationAngular app is running!');

  });

});

1. app.component.html

<div class="conatiner-fluid">

  <nav class="navbar navbar-expand-sm bg-dark navbar-dark">

    <ul class="navbar-nav">

      <li class="nav-item">

        <a routerLink="register-user" class="nav-link btn btn-primary" role="button">Registration</a>

      </li>

      <li class="nav-item">

        <a routerLink="login-user" class="nav-link btn btn-primary active" role="button">Login</a>

      </li>

    </ul>

  </nav>

<router-outlet></router-outlet>

</div>

1. app-routing.module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { RegisterUserComponent } from './components/register-user/register-user.component';

import { UserListComponent } from './components/user-list/user-list.component';

import { LoginUserComponent } from './components/login-user/login-user.component';

import { ActivateUserComponent } from './components/activate-user/activate-user.component';

import { AddBlogComponent } from './components/add-blog/add-blog.component';

import { ApproveBlogComponent } from './components/approve-blog/approve-blog.component';

import { ViewBlogComponent } from './components/view-blog/view-blog.component';

import { AdminHomeComponent } from './components/admin-home/admin-home.component';

const routes: Routes = [

  {path:'',redirectTo:'login-user', pathMatch:'full'},

  {path:'register-user', component:RegisterUserComponent},

  {path:'login-user', component:LoginUserComponent},

  {path:'user-list', component:UserListComponent},

  {path:'activate-user', component:ActivateUserComponent},

  {path:'add-blog', component:AddBlogComponent},

  {path:'approve-blog', component:ApproveBlogComponent},

  {path:'view-blog', component:ViewBlogComponent},

  {path:'admin-home/:id', component:AdminHomeComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

1. user.service.ts

import { Injectable } from '@angular/core';

import {HttpClient} from '@angular/common/http';

import {Observable} from 'rxjs';

import { User } from '../model/user';

@Injectable({

  providedIn: 'root'

})

export class UserService {

  private baseUrl='http://localhost:8080/api/';

  constructor(private http:HttpClient) { }

  getUserList():Observable<any>{

    return this.http.get(`${this.baseUrl}`+'user-list');

  }

  createUser(user: object):Observable<Object>{

    return this.http.post(`${this.baseUrl}`+'save-user',user);

  }

  deleteUser(userId: number):Observable<any>{

    return this.http.delete(`${this.baseUrl}/delete-user/${userId}`,{responseType:'text'});

  }

  getUser(userId: number):Observable<Object>{

    return this.http.get(`${this.baseUrl}/user/${userId}`);

  }

  updateUser(userId: number, value: any):Observable<Object>{

    return this.http.post(`${this.baseUrl}/update-user/${userId}`,value);

  }

  activateUser(userId: number):Observable<any>{

    return this.http.post(`${this.baseUrl}/active-user/${userId}`,{responseType:'text'});

  }

  validateUser(user:object):Observable<any>{

    return this.http.post(`${this.baseUrl}`+"validate-user",user);

  }

  logout(userId: number):Observable<any>{

    return this.http.post(`${this.baseUrl}/logout/${userId}`,{responseType:'text'});

  }

  }

1. user.service.spec.ts

import { TestBed } from '@angular/core/testing';

import { UserService } from './user.service';

describe('UserService', () => {

  let service: UserService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(UserService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

1. Create model as user.ts

export class User {

    userId:number;

    firstName:String;

    lastName:String;

    username:String;

    password:String;

    confirm\_password:String;

    email:String;

    role:String;

    status:String;

    isOnline:boolean;

    enabled:boolean;

}

1. User.spec.ts

import { User } from './user';

describe('User', () => {

  it('should create an instance', () => {

    expect(new User()).toBeTruthy();

  });

});

1. blog.ts

export class Blog {

    blogId:number;

    blogTitle:String;

    blogContent:String;

    blogPosted:String;

    status:String;

    noOfLikes:number;

    noOfViews:number;

    noOfComments:number;

    userId:number;

    username:String;

}

1. blog.spec.ts

import { Blog } from './blog';

describe('Blog', () => {

  it('should create an instance', () => {

    expect(new Blog()).toBeTruthy();

  });

});

1. register-user.component.ts

import { Component, OnInit } from '@angular/core';

import {UserService} from '../../services/user.service';

import {FormControl, FormGroup, Validators} from '@angular/forms';

import {User} from '../../model/user';

@Component({

  selector: 'app-register-user',

  templateUrl: './register-user.component.html',

  styleUrls: ['./register-user.component.scss']

})

export class RegisterUserComponent implements OnInit {

  user:User=new User();

  submitted=false;

  constructor(private userservice:UserService) { }

  ngOnInit(): void {

    this.submitted=false;

  }

  registrationform=new FormGroup({

    firstName:new FormControl('',[Validators.required]),

    lastName:new FormControl('',[Validators.required]),

    username:new FormControl('',[Validators.required]),

    password:new FormControl('',[Validators.required]),

    confirm\_password:new FormControl('',[Validators.required]),

    email:new FormControl('',[Validators.required, Validators.email]),

    role:new FormControl(),

  });

  register(register){

  this.user=new User();

    this.user.firstName=this.FirstName.value;

    this.user.lastName=this.LastName.value;

    this.user.username=this.Username.value;

    if(this.Password.value===this.ConfirmPassword.value)

      this.user.password=this.Password.value;

    this.user.email=this.Email.value;

    this.user.role=this.Role.value;

    if(this.user.role==="Admin"){

      this.user.enabled=true;

      this.user.status="Active";

    }

    else{

      this.user.enabled=false;

      this.user.status="Inactive";

    }

    this.user.isOnline=false;

    this.submitted=true;

    this.save();

  }

  save(){

    this.userservice.createUser(this.user)

    .subscribe(data=>console.log(data), error=>console.log(error));

    this.user=new User();

  }

  get FirstName(){

    return this.registrationform.get('firstName');

  }

  get LastName(){

    return this.registrationform.get('lastName');

  }

  get Username(){

    return this.registrationform.get('username');

  }

  get Password(){

    return this.registrationform.get('password');

  }

  get ConfirmPassword(){

    return this.registrationform.get('confirm\_password');

  }

  get Email(){

    return this.registrationform.get('email');

  }

  get Role(){

    return this.registrationform.get('role');

  }

  registrationForm(){

    this.submitted=false;

    this.registrationform.reset();

  }

}

1. register-user.component.spec.ts

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { RegisterUserComponent } from './register-user.component';

describe('RegisterUserComponent', () => {

  let component: RegisterUserComponent;

  let fixture: ComponentFixture<RegisterUserComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ RegisterUserComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(RegisterUserComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

1. register-user.component.html

<h3>Register Here</h3>

<div class="row">

    <div class="col-sm-4"></div>

    <div class="col-sm-4">

        <div [hidden]="submitted" style="width: 400px;">

            <form [formGroup]="registrationform" (ngSubmit)="register(register)">

                <div class="form-group">

                    <label for="firstName">First Name</label>

                    <input type="text" formControlName="firstName" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter First Name">

                    <div class="alert alert-danger" \*ngIf="(FirstName.touched) && (FirstName.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="FirstName.error.required">First Name is required</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="lastName">Last Name</label>

                    <input type="text" formControlName="lastName" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Last Name">

                    <div class="alert alert-danger" \*ngIf="(LastName.touched) && (LastName.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="LastName.error.required">Last Name is required</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="username">Username</label>

                    <input type="text" formControlName="username" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Username">

                    <div class="alert alert-danger" \*ngIf="(Username.touched) && (Username.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="Username.error.required">Username is required</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="password">Password</label>

                    <input type="password" formControlName="password" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Password">

                    <div class="alert alert-danger" \*ngIf="(Password.touched) && (Password.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="Password.error.required">Password is required</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="confirm\_password">Confirm Password</label>

                    <input type="password" formControlName="confirm\_password" class="form-control" data-toggle="tooltip" data-placement="right" title="Confirm Password" pattern="{{Password.value}}">

                    <div class="alert alert-danger" \*ngIf="(ConfirmPassword.touched) && (ConfirmPassword.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="ConfirmPassword.error.required">Confirm Password is required</span>

                        <span \*ngIf="ConfirmPassword.error.pattern">Password and Confirm Password does not match</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="email">Email</label>

                    <input type="text" formControlName="email" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Email Id">

                    <div class="alert alert-danger" \*ngIf="(Email.touched) && (Email.invalid)" style="margin-top: 5px;">

                        <span \*ngIf="Email.error.required">Email is required</span>

                        <span \*ngIf="Email.error.email">Invalid Email format</span>

                    </div>

                </div>

                <div class="form-group">

                    <label for="role">Role</label>

                    <select formControlName="role" class="form-control" data-toggle="tooltip" data-placement="right" title="Select user Role">

                        <option value="null">-- User Role --</option>

                        <option value="Admin">Admin</option>

                        <option value="User">User</option>

                    </select>

                </div>

                <button type="submit" class="btn btn-success">Submit</button>

            </form>

        </div>

    </div>

    <div>

        <div class="col-sm-4"></div>

    </div>

    <div class="col-sm-4">

        <div [hidden]="!submitted">

            <h4>Congratulations! You have registered successfully!</h4>

        </div>

    </div>

</div>

1. login-user.component.ts

import { Component, OnInit } from '@angular/core';

import {UserService} from '../../services/user.service';

import {FormControl, FormGroup,Validators} from '@angular/forms';

import {User} from'../../model/user';

import { data } from 'jquery';

import { Router } from '@angular/router';

@Component({

  selector: 'app-login-user',

  templateUrl: './login-user.component.html',

  styleUrls: ['./login-user.component.scss']

})

export class LoginUserComponent implements OnInit {

  user:User=new User();

  currentUser:any;

  submitted=false;

  constructor(private userservice:UserService , private router:Router) { }

  ngOnInit(): void {

    this.submitted=false;

  }

  loginform=new FormGroup({

    username:new FormControl('',[Validators.required]),

      password:new FormControl('',[Validators.required]),

    });

  login(login){

    this.user=new User();

    this.user.username=this.Username.value;

      this.user.password=this.Password.value;

      this.userservice.validateUser(this.user).subscribe(data=>{

        console.log(data);

        if(data!=null){

          this.currentUser=data;

          if(this.currentUser.role==="Admin" && this.currentUser.isOnline === true){

            this.router.navigateByUrl('/admin-home/' + `${this.currentUser.userId}`);

           }

           else{

             this.router.navigateByUrl('/user-home' + `${this.currentUser.userId}`);

           }

          console.log(this.currentUser.userId);

        }else {

          console.log("ObjectEmpty");

        }

      },error=>console.log(error));

  }

  get Username(){

    return this.loginform.get('username');

  }

  get Password(){

    return this.loginform.get('password');

  }

  }

1. login-user.component.spec.ts

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { LoginUserComponent } from './login-user.component';

describe('LoginUserComponent', () => {

  let component: LoginUserComponent;

  let fixture: ComponentFixture<LoginUserComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ LoginUserComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(LoginUserComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

1. login-user.component.html

<form action="action\_page.php" method="post">

    <div class="imgcontainer">

      <img src="img\_avatar2.png" alt="Avatar" class="avatar">

    </div>

    <form [formGroup]="loginform" (ngSubmit)="login(login)">

      <label for="uname"><b>Username</b></label>

      <input type="text" formControlName="username" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Username" placeholder="Enter Username">

      <div class="alert alert-danger" \*ngIf="(Username.touched) && (Username.invalid)">

        <span \*ngIf="Username.error.required">Username is required</span>

    </div>

    <label for="password">Password</label>

    <input type="password" formControlName="password" class="form-control" data-toggle="tooltip" data-placement="right" title="Enter Password" placeholder="Enter Password">

    <div class="alert alert-danger" \*ngIf="(Password.touched) && (Password.invalid)" style="margin-top: 5px;">

        <span \*ngIf="Password.error.required">Password is required</span>

    </div>

      <button type="submit">Login</button>

      <label>

        <input type="checkbox" checked="checked" name="remember"> Remember me

      </label>

    </form>

    <div class="container" style="background-color:#f1f1f1">

      <button type="button" class="cancelbtn">Cancel</button>

      <span class="psw">Forgot <a href="#">password?</a></span>

    </div>

  </form>

1. login-user.component.scss

/\* Bordered form \*/

form {

  border: 3px solid #f1f1f1;

}

/\* Full-width inputs \*/

input[type=text], input[type=password] {

  width: 100%;

  padding: 12px 20px;

  margin: 8px 0;

  display: inline-block;

  border: 1px solid #ccc;

  box-sizing: border-box;

}

/\* Set a style for all buttons \*/

button {

  background-color: #4CAF50;

  color: white;

  padding: 14px 20px;

  margin: 8px 0;

  border: none;

  cursor: pointer;

  width: 100%;

}

/\* Add a hover effect for buttons \*/

button:hover {

  opacity: 0.8;

}

/\* Extra style for the cancel button (red) \*/

.cancelbtn {

  width: auto;

  padding: 10px 18px;

  background-color: #f44336;

}

/\* Center the avatar image inside this container \*/

.imgcontainer {

  text-align: center;

  margin: 24px 0 12px 0;

}

/\* Avatar image \*/

img.avatar {

  width: 40%;

  border-radius: 50%;

}

/\* Add padding to containers \*/

.container {

  padding: 16px;

}

/\* The "Forgot password" text \*/

span.psw {

  float: right;

  padding-top: 16px;

}

/\* Change styles for span and cancel button on extra small screens \*/

@media screen and (max-width: 300px) {

  span.psw {

    display: block;

    float: none;

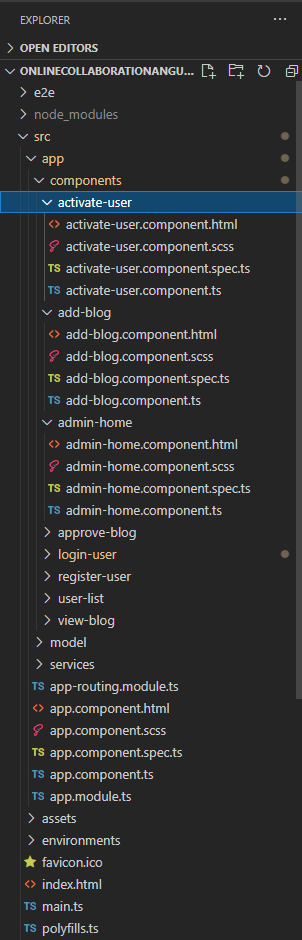
  }

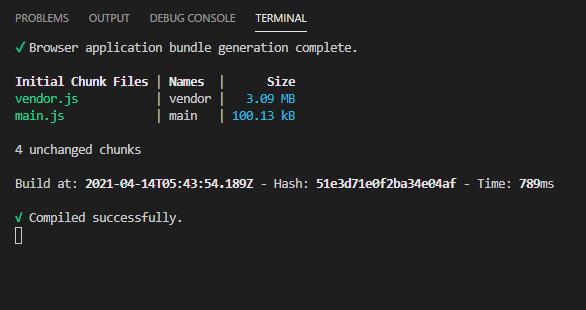
  .cancelbtn {

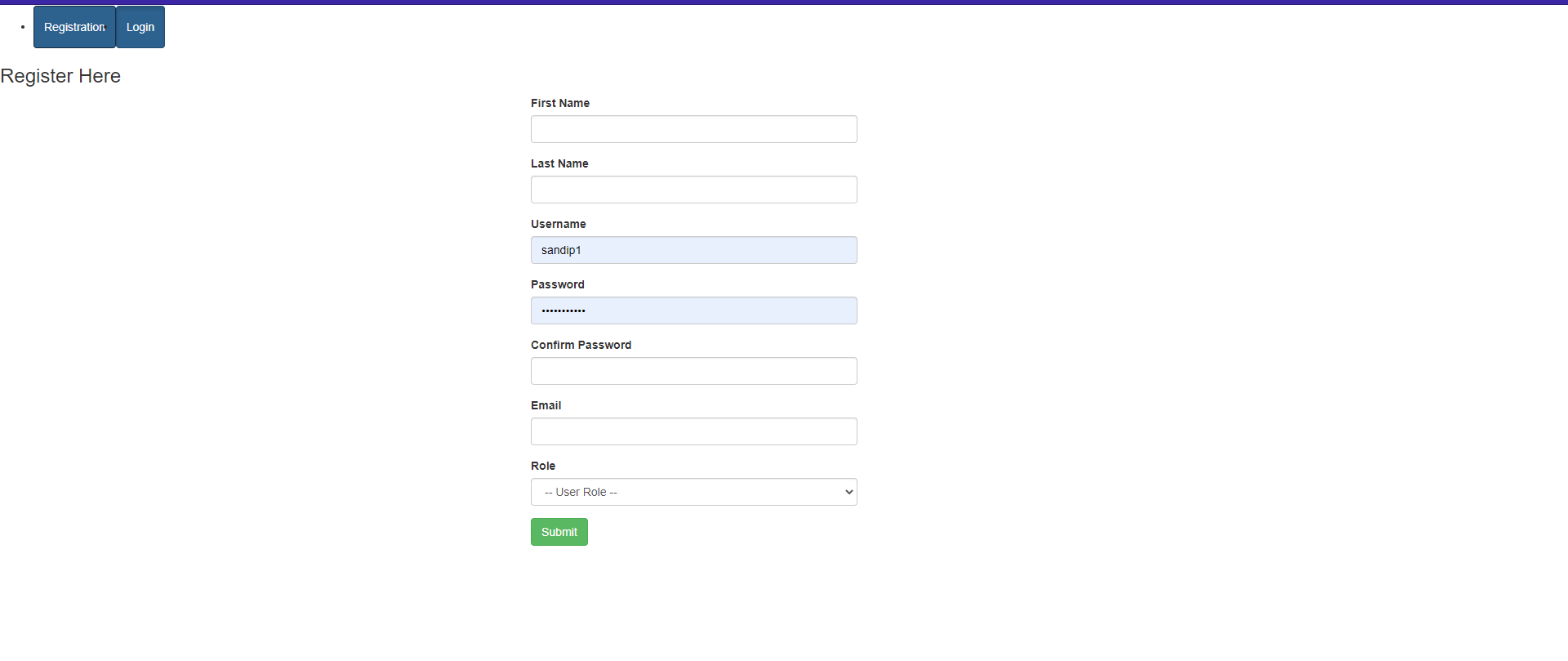
    width: 100%;

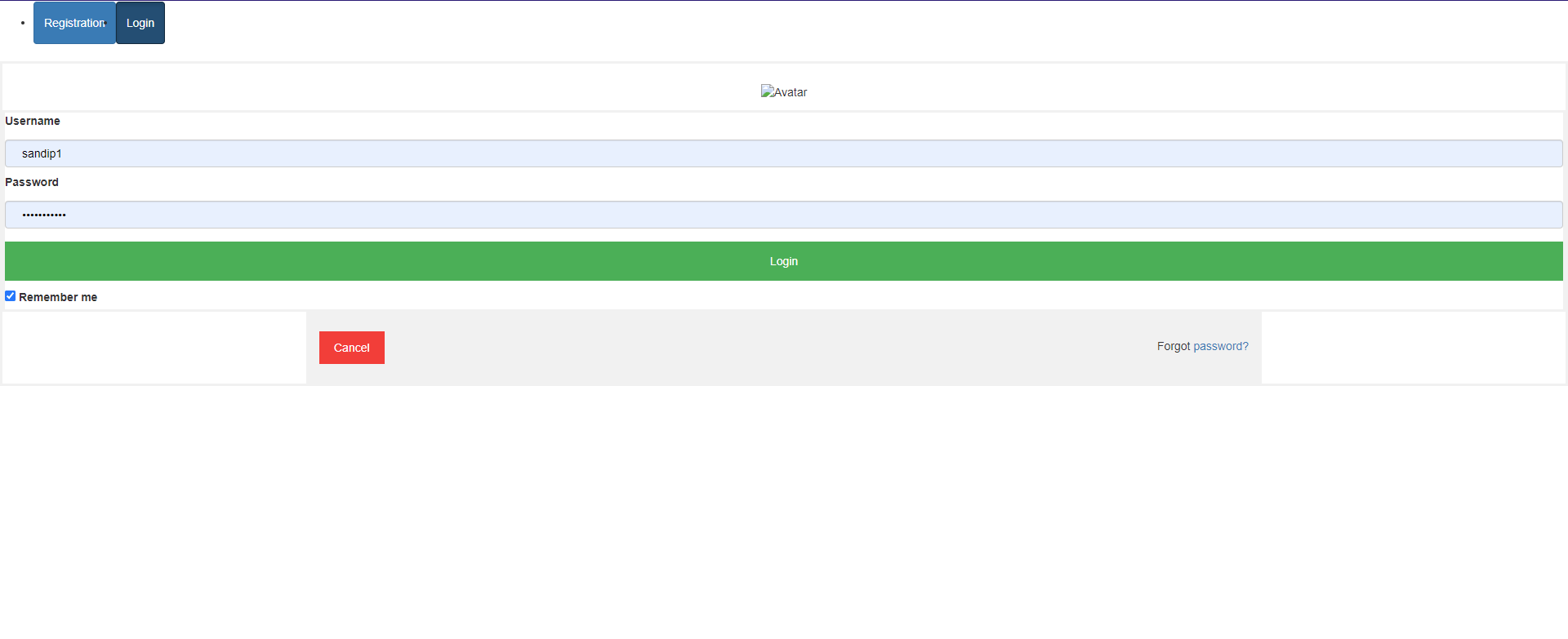
  }

}









References

1. <https://www.youtube.com/watch?v=7DzdebaSgxg&t=7937s>
2. <https://medium.com/javarevisited/a-simple-user-authentication-api-made-with-spring-boot-4a7135ff1eca>
3. <https://dzone.com/articles/add-login-to-your-spring-boot-app-in-10-mins>
4. <https://www.w3schools.com/howto/howto_css_login_form.asp>

