**MODEL PACKAGES**

**USER MODEL**

package com.alpha.Wellness\_Backend.model;

import java.io.Serializable;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Transient;

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;

@Transient

private String confirmPassword;

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 getConfirmPassword() {

return confirmPassword;

}

public void setConfirmPassword(String confirmPassword) {

this.confirmPassword = confirmPassword;

}

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;

}

}

**BLOG MODEL**

package com.alpha.Wellness\_Backend.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, noOfComments, 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 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;

}

}

BLOG COMMENTS MODEL

package com.alpha.Wellness\_Backend.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.JsonManagedReference;

@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 ProfileId;

String title;

int noOfLikes;

String blogComment;

LocalDate currentDate;

@ManyToOne

@JoinColumn(name="BlogId")

@JsonManagedReference

Blog blog;

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 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 getProfileId() {

return ProfileId;

}

public void setProfileId(String profileId) {

ProfileId = profileId;

}

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 static long getSerialversionuid() {

return serialVersionUID;

}

}

**DOMAIN RESPONSE MODEL**

**package** com.alpha.Wellness\_Backend.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;

}

}

**JOB APPLIED MODEL**

package com.alpha.Wellness\_Backend.model;

import java.io.Serializable;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

public class Job implements Serializable{

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

private int jobId;

private String jobName;

private String jobDescription;

private String jobLocation;

private boolean isActive;

public int getJobId() {

return jobId;

}

public void setJobId(int jobId) {

this.jobId = jobId;

}

public String getJobName() {

return jobName;

}

public void setJobName(String jobName) {

this.jobName = jobName;

}

public String getJobDescription() {

return jobDescription;

}

public void setJobDescription(String jobDescription) {

this.jobDescription = jobDescription;

}

public String getJobLocation() {

return jobLocation;

}

public void setJobLocation(String jobLocation) {

this.jobLocation = jobLocation;

}

public boolean isActive() {

return isActive;

}

public void setActive(boolean isActive) {

this.isActive = isActive;

}

public static long getSerialversionuid() {

return serialVersionUID;

}

}

**SERVICE PACKAGE**

**IUserService**

package com.alpha.Wellness\_Backend.service;

import java.util.List;

import com.alpha.Wellness\_Backend.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 deactivateUser(int userId);

boolean updateUserProfile(String file, Integer userId);

}

**IBlogService**

package com.alpha.Wellness\_Backend.service;

import java.util.List;

import com.alpha.Wellness\_Backend.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);

}

**IBlogCommentsService**

package com.alpha.Wellness\_Backend.service;

import java.util.List;

import com.alpha.Wellness\_Backend.model.BlogComments;

public interface IBlogCommentsService {

List<BlogComments> getAllBlogsComments();

BlogComments getBlogCommentsById(int blogCommentsId);

boolean addBlogComments(BlogComments blogComments);

boolean updateBlogComments(BlogComments blogComments);

boolean deleteBlogComments(BlogComments blogComments);

}

**IJobService**

package com.alpha.Wellness\_Backend.service;

import java.util.List;

import com.alpha.Wellness\_Backend.model.Job;

public interface IJobService {

List<Job> getAllJobs();

Job getJobById(int jobId);

Job getJobByName(String jobName);

boolean addJob(Job job);

boolean updateJob(Job job);

boolean deleteJob(Job job);

boolean deactivateJob(int jobId);

}

**ServiceImpl Package**

**UserServiceImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IUserDao;

import com.alpha.Wellness\_Backend.model.User;

import com.alpha.Wellness\_Backend.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 deactivateUser(int userId) {

return userDao.deactivateUser(userId);

}

@Override

public boolean updateUserProfile(String file, Integer userId) {

return userDao.updateUserProfile(file, userId);

}

}

**BlogServiceImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IBlogDao;

import com.alpha.Wellness\_Backend.model.Blog;

import com.alpha.Wellness\_Backend.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);

}

}

**BlogCommentsServiceImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IBlogCommentsDao;

import com.alpha.Wellness\_Backend.model.BlogComments;

import com.alpha.Wellness\_Backend.service.IBlogCommentsService;

@Service

@Transactional

public class BlogCommentsServiceImpl implements IBlogCommentsService{

@Autowired

IBlogCommentsDao blogCommentsDao;

@Override

public List<BlogComments> getAllBlogsComments() {

return blogCommentsDao.getAllBlogsComments();

}

@Override

public BlogComments getBlogCommentsById(int blogCommentsId) {

return blogCommentsDao.getBlogCommentsById(blogCommentsId);

}

@Override

public boolean addBlogComments(BlogComments blogComments) {

return blogCommentsDao.addBlogComments(blogComments);

}

@Override

public boolean updateBlogComments(BlogComments blogComments) {

return blogCommentsDao.updateBlogComments(blogComments);

}

@Override

public boolean deleteBlogComments(BlogComments blogComments) {

return blogCommentsDao.deleteBlogComments(blogComments);

}

}

**JobServiceImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IJobDao;

import com.alpha.Wellness\_Backend.model.Job;

import com.alpha.Wellness\_Backend.service.IJobService;

@Service

@Transactional

public class JobServiceImpl implements IJobService{

@Autowired

IJobDao jobDao;

@Override

public List<Job> getAllJobs() {

return jobDao.getAllJobs();

}

@Override

public Job getJobById(int jobId) {

return jobDao.getJobById(jobId);

}

@Override

public Job getJobByName(String jobName) {

return jobDao.getJobByName(jobName);

}

@Override

public boolean addJob(Job job) {

return jobDao.addJob(job);

}

@Override

public boolean updateJob(Job job) {

return jobDao.updateJob(job);

}

@Override

public boolean deleteJob(Job job) {

return jobDao.deleteJob(job);

}

@Override

public boolean deactivateJob(int jobId) {

return jobDao.deactivateJob(jobId);

}

}

**Data Access Object model**

**IUserDao**

package com.alpha.Wellness\_Backend.dao;

import java.util.List;

import com.alpha.Wellness\_Backend.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);

boolean updateUserProfile(String file, Integer userId);

}

**IBlogDao**

package com.alpha.Wellness\_Backend.dao;

import java.util.List;

import com.alpha.Wellness\_Backend.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);

}

**IBlogCommentsDao**

package com.alpha.Wellness\_Backend.dao;

import java.util.List;

import com.alpha.Wellness\_Backend.model.BlogComments;

public interface IBlogCommentsDao {

List<BlogComments> getAllBlogsComments();

BlogComments getBlogCommentsById(int blogCommentsId);

boolean addBlogComments(BlogComments blogComments);

boolean updateBlogComments(BlogComments blogComments);

boolean deleteBlogComments(BlogComments blogComments);

}

**IJobDao**

package com.alpha.Wellness\_Backend.dao;

import java.util.List;

import com.alpha.Wellness\_Backend.model.Job;

public interface IJobDao {

List<Job> getAllJobs();

Job getJobById(int jobId);

Job getJobByName(String jobName);

boolean addJob(Job job);

boolean updateJob(Job job);

boolean deleteJob(Job job);

boolean deactivateJob(int jobId);

}

**Dao Impl**

**UserDaoImpl**

package com.alpha.Wellness\_Backend.daoImpl;

import java.util.List;

import org.hibernate.SessionFactory;

import org.hibernate.query.Query;

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

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.alpha.Wellness\_Backend.dao.IUserDao;

import com.alpha.Wellness\_Backend.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();

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 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;

}

}

}

**BlogDaoImpl**

package com.alpha.Wellness\_Backend.daoImpl;

import java.util.List;

import javax.persistence.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.alpha.Wellness\_Backend.dao.IBlogDao;

import com.alpha.Wellness\_Backend.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 id) {

// String q="select blogTitle, blogContent from Blog where blogId='"+id+"'";

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

// return query.getResultList();

return null;

}

@Override

public List<Blog> mainList() {

// TODO Auto-generated method stub

return null;

}

@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;

}

}

}

**BlogCommentsDaoImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IBlogCommentsDao;

import com.alpha.Wellness\_Backend.model.BlogComments;

@Repository("blogCommentsDao")

@Transactional

public class BlogCommentsDaoImpl implements IBlogCommentsDao{

@Autowired

SessionFactory sessionFactory;

@Override

public List<BlogComments> getAllBlogsComments() {

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

}

@Override

public BlogComments getBlogCommentsById(int blogCommentsId) {

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

}

@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;

}

}

}

**JobDaoImpl**

package com.alpha.Wellness\_Backend.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.alpha.Wellness\_Backend.dao.IJobDao;

import com.alpha.Wellness\_Backend.model.Job;

@Repository("jobDao")

@Transactional

public class JobDaoImpl implements IJobDao{

@Autowired

SessionFactory sessionFactory;

@Override

public List<Job> getAllJobs() {

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

}

@Override

public Job getJobById(int jobId) {

return sessionFactory.getCurrentSession().get(Job.class, Integer.valueOf(jobId));

}

@Override

public Job getJobByName(String jobName) {

String query="from job where jobname=:jobname";

return sessionFactory.getCurrentSession().createQuery(query, Job.class).setParameter("jobname", jobName).getSingleResult();

}

@Override

public boolean addJob(Job job) {

try {

sessionFactory.getCurrentSession().save(job);

return true;

}

catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateJob(Job job) {

try {

sessionFactory.getCurrentSession().update(job);

return true;

}

catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteJob(Job job) {

try {

sessionFactory.getCurrentSession().delete(job);

return true;

}

catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean deactivateJob(int jobId) {

try {

Job job=getJobById(jobId);

job.setActive(false);

sessionFactory.getCurrentSession().update(job);

return true;

}

catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

}

**Controller**

**UserController**

package com.alpha.Wellness\_Backend.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.alpha.Wellness\_Backend.model.User;

import com.alpha.Wellness\_Backend.service.IUserService;

@RestController

@CrossOrigin(origins="http://localhost:4200")

@RequestMapping(value="/api")

public class UserController {

@Autowired

IUserService userService;

@PostMapping("save-user")

public boolean saveUser (@RequestBody User user) {

return userService.addUser(user);

}

@GetMapping("user-list")

public List<User> allUsers() {

return userService.getAllUsers();

}

@DeleteMapping("delete-user/{user\_id}")

public boolean deleteUser(@PathVariable("user\_id") int user\_id) {

return userService.deleteUser(user\_id);

}

@GetMapping("user/{user\_id}")

public User userById(@PathVariable("user\_id") int user\_id) {

return userService.getUserById(user\_id);

}

@PostMapping("update-user/{user\_id}")

public boolean updateUser(@RequestBody User user, @PathVariable("user\_id") int user\_id) {

user.setUserId(user\_id);

return userService.updateUser(user);

}

}

**BlogController**

package com.alpha.Wellness\_Backend.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.alpha.Wellness\_Backend.model.Blog;

import com.alpha.Wellness\_Backend.model.User;

import com.alpha.Wellness\_Backend.service.IBlogService;

@RestController

@CrossOrigin(origins="http://localhost:4200")

@RequestMapping(value="/api")

public class BlogController {

@Autowired

IBlogService blogService;

@PostMapping("save-blog")

public boolean saveBlog(@RequestBody Blog blog ) {

return blogService.addBlog(blog);

}

@PostMapping("blog-list")

public List<Blog> allBlogs() {

return blogService.getAllBlogs();

}

@DeleteMapping("delete-blog/{blog}")

public boolean deleteUser(@PathVariable("blog") Blog blog) {

return blogService.deleteBlog(blog);

}

@GetMapping("blog/{blogId}")

public Blog blogById(@PathVariable("blogId") int blogId) {

return blogService.getBlogById(blogId);

}

@PostMapping("update-blog/{blogId}")

public boolean updateBlog(@RequestBody Blog blog, @PathVariable("blogId") int blogId) {

blog.setBlogId(blogId);

return blogService.updateBlog(blog);

}

}

**JobController**

package com.alpha.Wellness\_Backend.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.alpha.Wellness\_Backend.model.Job;

import com.alpha.Wellness\_Backend.service.IJobService;

@RestController

@CrossOrigin(origins = "http://localhost:4200")

@RequestMapping(value="/api")

public class JobController {

@Autowired

IJobService jobService;

@PostMapping("save-job")

public boolean saveJob (@RequestBody Job job) {

return jobService.addJob(job);

}

@GetMapping("job-list")

public List<Job> allJobs() {

return jobService.getAllJobs();

}

@DeleteMapping("delete-job/{job\_id}")

public boolean deleteJob(@PathVariable("job\_id") int job\_id) {

Job job = jobService.getJobById(job\_id);

return jobService.deleteJob(job);

}

@GetMapping("job/{job\_id}")

public Job jobById(@PathVariable("job\_id") int job\_id) {

return jobService.getJobById(job\_id);

}

@PostMapping ("update-job/{job\_id}")

public boolean updateJob(@RequestBody Job job, @PathVariable("job\_id") int job\_id) {

job.setJobId(job\_id);

return jobService.updateJob(job);

}

}