App Config:

```
package klu.config;

import
org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherS
ervletInitializer;

public class AppConfig extends
AbstractAnnotationConfigDispatcherServletInitializer {

	@Override
	protected Class<?>[] getRootConfigClasses() {
		// TODO Auto-generated method stub
		return new Class[] {MvcConfig.class};
	}

	@Override
	protected Class<?>[] getServletConfigClasses() {
		// TODO Auto-generated method stub
		return null;
	}

	@Override
	protected String[] getServletMappings() {
		// TODO Auto-generated method stub
		return new String[] {"/"};
	}

}
```

MvcConfig:

```
package klu.config;


import javax.sql.DataSource;


import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;

import org.springframework.web.servlet.config.annotation.ViewResolverRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;


@Configuration

@ComponentScan("klu")

@EnableWebMvc
```

```
public class MvcConfig implements WebMvcConfigurer  {
	@Override
	public void configureViewResolvers(ViewResolverRegistry registry) {
		registry.jsp("/",".jsp");
	}

	@Bean
	public DataSource dataSource() {
		DriverManagerDataSource DS = new DriverManagerDataSource();
		DS.setDriverClassName("com.mysql.cj.jdbc.Driver");
		DS.setUrl("jdbc:mysql://localhost:3306/klu");
		DS.setUsername("root");
		DS.setPassword("Ep@klu123");
		return DS;
	}

	@Bean
	public JdbcTemplate jdbcTemplate(DataSource DS) {
		return new JdbcTemplate(DS);
	}
}

package klu.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class AppController {
	@RequestMapping("/")
	public ModelAndView index()
	{
```

```
		ModelAndView MV = new ModelAndView();
		MV.setViewName("student");
		return MV;
	}
}
package klu.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

import klu.model.Student;
import klu.model.StudentManager;

@Controller
@ResponseBody
@RequestMapping("/student")
public class StudentController {

	@Autowired
	StudentManager SM;

	@PostMapping("/save")
	public String save(@RequestBody Student S)
	{
		return SM.saveData(S);
```

```
	}

	@PutMapping("/update")
	public String update(@RequestBody Student S)
	{
		return SM.updateData(S);
	}

	@DeleteMapping("/delete")
	public String delete(@RequestParam("id") int id)
	{
		return SM.deleteData(id);
	}

	@GetMapping("/read")
	public String read()
	{
		return SM.readData().toString();
	}

	@GetMapping("/readjson")
	public String readjson()
	{
		return SM.readJSONData().toString();
	}
}
```

```
package klu.model;

public class Student {
	int id;
	String name;
	String dept;
	public Student(int id, String name, String dept)
	{
		this.id = id;
		this.name = name;
		this.dept = dept;
```

```
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getDept() {
		return dept;
	}
	public void setDept(String dept) {
		this.dept = dept;
	}
	@Override
	public String toString() {
		return "Student [id=" + id + ", name=" + name + ", dept=" +
dept + "]";
	}
}
```

```
package klu.model;


import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;


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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.stereotype.Service;


import com.google.gson.Gson;

import com.google.gson.GsonBuilder;


@Service

public class StudentManager {


	JdbcTemplate jdbcTemplate;

	//Context Dependency Injections (Constructor Method instead of Setter Method)
```

```
	@Autowired
	public StudentManager(JdbcTemplate jdbcTemplate)
	{
		this.jdbcTemplate = jdbcTemplate;
	}

	//INSERT OPERATION
	public String saveData(Student S)
	{
		try
		{
			String qry = "insert into student values(" + S.getId() + ", '"+ S.getName() +"', '" + S.getDept() + "')";
			jdbcTemplate.update(qry); //INSERT
			return "New student has been added";
		}catch(Exception e)
		{
			return e.getMessage();
		}
	}

	//UPDATE OPERATION
	public String updateData(Student S)
	{
		try
		{
			String qry = "update student set name='"+ S.getName() +"',dept='"+ S.getDept() +"' where id='"+ S.getId() +"'";
			jdbcTemplate.update(qry);
			return "Student details has been updated";
		}catch (Exception e)
		{
			return e.getMessage();
		}
```

```
}

//DELETE OPERATION
public String deleteData(int id)
{
	try
	{
		String qry = "delete from student where id="+ id +"";
		jdbcTemplate.update(qry);
		return "Student data has been deleted";
	}catch(Exception e)
	{
		return e.getMessage();
	}
}

//READ OPERATION
public List<Student> readData()
{
	List<Student> slist = new ArrayList<Student>();
	String qry = "select * from student";
	slist = jdbcTemplate.query(qry, new RowMapper<Student>() {

		@Override
		public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
			Student tmp = new Student(rs.getInt(1) , rs.getString(2) , rs.getString(3) );
			return tmp;
		}

	});
	return slist;
}
```

```
	//READ OPERATION with RETURN JSON
		public List<String> readJSONData()
		{
			List<Student> slist = new ArrayList<Student>();
			String qry = "select * from student";
			slist = jdbcTemplate.query(qry, new RowMapper<Student>() {

				@Override
				public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
					Student tmp = new Student(rs.getInt(1) , rs.getString(2) ,
rs.getString(3) );
					return tmp;
				}

			});

			List<String> result = new ArrayList<String>();
			for(Student S:slist)
				result.add(toJSON(S));
			return result;
		}

	//Convert Java object to JSON
	public String toJSON(Object obj) {
		Gson G = new GsonBuilder().create();
		return G.toJson(obj);
	}
}
```

Name:P.UdayKumar
ID:2200031172