1)

a) Spring Framework: Spring açık kaynak kodlu bir framework'tür. Bünyesinde Spring JDBC, Spring MVC, Spring Security, Spring AOP, Spring ORM, Spring Test gibi modülleri içerir. Spring Framework IoC (Inversion of Control) yazılım tasarım prensibini temel alır, Dependency Injection yapılmasını sağlar. Web uygulamaların geliştirilmesinde ve mikroservis yazılımlarında kullanılabilir.[1]

Spring Boot ise Spring'i temel alan bir framework'tür.

Spring Boot kod parçası örneği;

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class HelloWorldApplication {


    public static void main(String[] args) {

        SpringApplication.run(HelloWorldApplication.class, args);

    }

}
```

b) Hibernate: Hibernate, veritabanı işlemlerinde kullanılan bir framework'tür. ORM (Object Relational Mapping) aracı olarak kullanılır. ORM, uygulamamızdaki nesnelerle ilişkisel veritabanları arasında bağ kurmamızı sağlayan bir yapıdır. Yerel SQL sorguları yazmadan verileri saklamayı, almayı ve değiştirmeyi kolaylaştırır.

Örnek; [2]

```
@Entity

@Table(name = "employees")

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String name;

    // Other properties, getters, and setters

}


Session session = sessionFactory.openSession();
```

```
Transaction tx = session.beginTransaction();

List<Employee> employees = session.createQuery("FROM Employee",
Employee.class).list();

tx.commit();

session.close();
```

c) Apache Struts: Apache Struts, kurumsal kullanıma hazır web uygulamaları oluşturmak için MVC (Model View Controller) mimarisini kullanan bir framework'tür. [3]

Kod örneği; [4]

```
package org.apache.struts.helloworld.action;

import org.apache.struts.helloworld.model.MessageStore;

import com.opensymphony.xwork2.ActionSupport;

public class HelloWorldAction extends ActionSupport {

    private MessageStore messageStore;

    public String execute() {

        messageStore = new MessageStore() ;

        return SUCCESS;

    }

    public MessageStore getMessageStore() {

        return messageStore;

    }

}
```

d) JUnit: JUnit, Java uygulamaları için kullanılan bir test framework'üdür. Geliştiricilerin kodlarının doğruluğunu sağlamak için birim testleri yazmasına olanak tanır.

Örnek; [2]

```
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class MathUtilsTest {

    @Test

    public void testAddition() {

        int result = MathUtils.add(5, 10);

        assertEquals(15, result);

    }
```

```
}
```

e) Vaadin: Vaadin Flow (Vaadin Framework), web uygulamaları ve web siteleri oluşturmaya yönelik bir Java web framework'üdür. Doğrudan HTML veya JavaScript kullanılmasına gerek kalmadan kullanıcı arayüzlerinin, Java kullanılarak oluşturulmasına olanak tanır.

Örnek kod parçası; [5]

```
@Route("hello-world") // exposes the view through
http://localhost:8080/hello-world
public class MainView extends VerticalLayout {
    public MainView() {
        // creates a text field
        TextField textField = new TextField("Enter your name");
        // creates a button
        Button button = new Button("Send");
        // adds behaviour to the button using the click event
        button.addClickListener(event ->
                add(new Paragraph("Hello, " + textField.getValue()))
        );
        // adds the UI components to the view (VerticalLayout)
        add(textField, button);
    }
}
```

f) Play Framework: Play, MVC mimarisini benimseyen bir web uygulaması framework'üdür. RESTful yapıdadır. [6]

Aşağıdaki kod parçası bir controller sınıfını içerir; [7]

```
import play.mvc.*;
public class HomeController extends Controller {
    public Result index() {
        return ok(views.html.index.render());
    }
    public Result explore() {
        return ok(views.html.explore.render());
    }
    public Result tutorial() {
```

```
        return ok(views.html.tutorial.render());
    }
}
```

g) ZK: ZK, Java kullanılarak yazılmıştır. Web uygulamaları için grafiksel kullanıcı arayüzlerinin oluşturulmasına olanak tanıyan açık kaynaklı bir Ajax Web uygulaması framework'üdür.

Aşağıda bir checkBox elementinin işaretlenmesi sonucunda aktifleşen buton örneği görülmektedir. [8]

```
import org.zkoss.zk.ui.Component;
import org.zkoss.zk.ui.select.SelectorComposer;
import org.zkoss.zk.ui.select.annotation.Listen;
import org.zkoss.zk.ui.select.annotation.Wire;
import org.zkoss.zul.Button;
import org.zkoss.zul.Checkbox;
public class RegistrationComposer extends
SelectorComposer<Component> {
     @Wire
     private Button submitButton;
     @Wire
     private Checkbox acceptTermBox;

     @Listen("onCheck = #acceptTermBox")
     public void changeSubmitStatus(){
          if (acceptTermBox.isChecked()){
               submitButton.setDisabled(false);
               submitButton.setImage("/images/submit.png");
          }else{
               submitButton.setDisabled(true);
               submitButton.setImage("");
          }
     }
}
```

2) SOA (Service-Oriented Architecture), yani hizmet odaklı mimari, bir yazılım mimarisi yaklaşımıdır. Yeniden kullanılabilir hizmetlerin birbirleriyle bağlantılar kurularak birlikte çalışmasının sağlanıldığı mimari yapıdır. Örneğin; ödeme işlemi için her bir sistem için yeniden yazmak yerine, ödeme işlemi için tek bir defa yazılan sistem birden fazla yazılımla entegre edilerek kullanılabilir. [9]

Web servis, elektronik cihaz tarafından başka bir elektronik cihaza sunulan, World Wide Web üzerinden birbirleriyle iletişim kuran yapıların bütününe verilen isimdir. [10] İnternet üzerinde bulunan herhangi bir web sitesi, web servisi olarak çalışır. RESTful servisler, Representational State Transfer (REST) prensiplerine uygun olarak tasarlanmış ve uygulanmış web servisleridir. Bu servisler, web üzerinden kaynaklara (resources) erişim sağlayan, bu kaynaklar üzerinde işlemler gerçekleştiren ve genellikle JSON veya XML gibi veri formatlarını kullanarak iletişim kuran servislerdir. [11] Örneğin, bir kullanıcı kaydı oluşturmak, kullanıcı bilgilerini almak, güncellemek veya silmek için bir RESTful API tasarlayabiliriz. Bu API, HTTP yöntemlerini (GET, POST, PUT, DELETE) kullanarak kullanıcı kayıtlarını yönetebilir.

HTTP (Hypertext Transfer Protocol), web üzerinde iletişim kurmak için kullanılan bir iletişim protokolüdür. HTTP'nin temel amacı, istemci ve sunucu arasında veri alışverişi yapmaktır. Bu alışveriş sırasında, çeşitli HTTP yöntemleri veya metotları kullanılır. Temel HTTP metotları:

GET: Sunucudan belirli bir kaynağın (örneğin, bir web sayfası, bir resim veya bir dosya) bilgisini istemek için kullanılır.

POST: Sunucuya yeni bir kaynak oluşturmak veya mevcut bir kaynağı güncellemek için kullanılır.

PUT: Sunucuya belirli bir kaynağı oluşturmak veya güncellemek için kullanılır.

DELETE: Sunucudan belirli bir kaynağı silmek için kullanılır.

PATCH: Sunucuda belirli bir kaynağı kısmen güncellemek için kullanılır. PATCH metodu, belirtilen kaynağın yalnızca belirtilen veriyle kısmi olarak değiştirilmesini gerektirir. [12]

KAYNAKÇA

[1] https://www.turing.com/kb/spring-vs-spring-boots-best-web-apps

[2] https://medium.com/thefreshwrites/exploring-the-most-common-java-libraries-frameworks-6c7ae547d23c

[3] https://www.jrebel.com/blog/apache-struts

[4] https://struts.apache.org/getting-started/hello-world-using-struts2

[5] https://en.wikipedia.org/wiki/Vaadin

[6] https://www.playframework.com/

[7] https://github.com/playframework/play-samples/blob/3.0.x/play-java-hello-world-tutorial/app/controllers/HomeController.java

[8] https://www.zkoss.org/wiki/ZK_Getting_Started/Learn_ZK_in_10_Minutes

[9] https://aws.amazon.com/tr/what-is/service-oriented-architecture/

[10] https://tr.wikipedia.org/wiki/Web_servis

[11] https://medium.com/@ilkebasalak/web-servis-soap-ve-rest-nedir-8e7e03f28a9b

[12] https://www.geeksforgeeks.org/different-kinds-of-http-requests/