THE STATE UNIVERSITY OF ZANZIBAR (SUZA)

SCHOOL OF COMPUTING COMMUNICATION AND MEDIA STUDIES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATIONTECHNOLOGY

STUDENT'S NAME: **ALI HAMAD RASHID**

REGISTRATION NUMBER: **BITAM/10/22/009/TZ**

COURSE NAME: **IS DEVELOPMENT FRAMEWORKS AND METHODS**

COURSE CODE: **INF 2108**

INSTRUCTOR'S NAME: **DR. MARYAM**

TYPE OF ASSIGNMENT: **INDIVIDUAL ASSIGNMENT**

SUBMISION DATE: **24/04/2024**

1. **DJANGO:** A high-level Python framework known for its batteries-included approach, emphasizing rapid development, security, and scalability.

Benefit of the framework: i) Faster development ii) Robust security features iii) Scalability.

Language used: Language that used in this frame work is Python.

Architecture of each framework: The architecture of this frame work is Model-View-Controller (MVC) although Django developers often refer as (MVT) where “T” stand for templates. Model is the structure of storing the data in the database, the view is a python function used to handle the web request, and the template contains static content like HTML, CSS, and Java Script.

The license for each framework: Django is released under a permissive open-source license known as the "BSD license" (Berkeley Software Distribution). This license allows you to use, modify, and distribute Django freely, both for commercial and non-commercial purposes.

2. **LARAVEL:** A PHP framework that provides a clean and elegant syntax, along with a rich set of features for web application development. It follows the MVC pattern as well.

Benefit of the framework: i) Simple website maintenance ii) Expressive syntax iii) Rich feature set

Language used by each framework: Language that used in this frame work is PHP.

Architecture of each framework: follows the Model-View-Controller (MVC) architectural pattern. Model in Laravel, represent the data structures of the application. View is the View layer in Laravel is responsible for rendering the user interface. Controllers in Laravel handle user requests and contain the application's business logic.

License for each framework: Laravel is released under the open-source MIT License. The MIT License is a permissive license that allows developers to use, modify, and distribute the software for any purpose, including commercial use, without requiring them to share their changes.

3. **ANGULAR**: A Type Script-based framework maintained by Google, used for building large-scale and feature-rich web applications. It follows the Component-Based Architecture.

Benefit of the Angular framework

I) Large-scale application development ii) Powerful features iii) Excellent Type Script support.

Language used by Angular framework: Language that used in this frame work is Type Script

Architecture of Angular framework: Angular use Component-Based Architecture to emphasizes the composition of applications using reusable and modular components, with features like templates, directives, services, modules to create dynamic, interactive web applications.

License for each framework: Angular is released under the open-source MIT License. This license allows developers to use, modify, and distribute Angular for any purpose.

4. **REACT:** A JavaScript library developed by Facebook, used for building user interfaces.

Benefit of the framework

i) Component reusability ii) Virtual DOM for efficient rendering iii) Community support.

Language used by React framework: Language that used in this frame work is JavaScript.

Architecture of React framework: React architecture revolves around the principles of component-based development, virtual DOM reconciliation, and a unidirectional data flow model. This architecture enables developers to build interactive and scalable user interfaces.

License for react framework: React developed and maintained by Facebook, is released under the MIT License. This license permits users to utilize, modify, and distribute React.js.

5. **SPRING BOOT:** A Java-based framework that simplifies the development of Java applications, including web applications.

Benefit of the framework

i) Rapid development ii) Dependency management iii) Enterprise-grade features.

Language used by each framework: Language that used in this frame work is Java

Architecture of Spring Boot framework: Spring Boot primarily follows the Model-View-Controller (MVC) architectural pattern. Model represents the data and business logic of the application. View represents the presentation layer of the application, responsible for rendering the user interface. Controller acts as an intermediary between the model and the view, handling user requests and coordinating the flow of data between the two.

License for spring boot framework: Spring Boot is licensed under the Apache License 2.0. This is a permissive open-source license that allows users to use, modify, and distribute the software for any purpose, both commercially and non-commercially.

6. **RUBY ON RETAIL:** A Ruby-based framework that follows the Convention over Configuration (CoC) principle, offering a productive and elegant development experience.

Benefit of Ruby on Rails

i) Productivity ii) convention-driven development iii) extensive libraries/gems.

Language of ruby on retail framework: Language that used in this frame work is Ruby

Architecture of Ruby on Rails framework: Model-View-Controller (MVC)

The license for each framework: Ruby on Rails use MIT architecture

7. **EXPRESS.JS:** A minimalist and flexible framework that allows building web applications and APIs quickly.

Benefit of Express.js

i) Lightweight ii) flexibility iii) easy integration with other Node.js modules.

Language of ruby on retail framework: Language that used in this frame work is JavaScript.

Architecture of Ruby on Rails framework: Model-View-Controller (MVC) or Model-View-Router (MVR)

License of express.js framework: express.js use MIT architecture

8. **FLASK:** A lightweight Python micro-framework that focuses on simplicity and extensibility.

Benefit of Flask i) Simplicity ii) extensibility iii) minimalistic approach.

Language of ruby on retail framework: Language that used in this frame work is Python

Architecture of Ruby on Rails framework: Model-View-Controller (MVC).

License of express.js framework: Flask use BSD.

9. **VUE.JS:** A progressive JavaScript framework for building user interfaces.

Benefit of Vue.js i) Simplicity ii) versatility iii) smooth learning curve.

Language of Vue framework: Language that used in this frame work is JavaScript

Architecture of Vue framework: Component-Based Architecture.

License of Vue framework: Views use MIT architecture.

10. **ASP.NET:** A framework developed by Microsoft, it supports multiple programming languages such as C#.

Benefit of ASP.NET: i) Cross-platform compatibility ii) high-performance iii) modular architecture.

Language of ASP.NET framework: Language that used in this frame work is C#.

Architecture of ASP.NET framework: Varies depending on the version (e.g., MVC etc.)

License of ASP.NET framework: Apache License 2.0

11. **EMBER.JS:** A JavaScript framework for building ambitious web applications.

Benefit of Ember.js

i) Ambitious application development ii) convention over configuration iii) scalability.

Language of Ember.js framework: Language that used in this frame work is JavaScript

Architecture of Ember.js framework: Convention over Configuration (CoC)

License of Ember.js framework: MIT.

12. **ASP.NET Core:** A cross-platform, open-source framework for building modern web applications with.

Benefit of ASP.NET Core i) Cross-platform compatibility ii) high-performance iii) modular architecture.

The language of ASP.NET Core: The language used by ASP.NET core C# (supports multiple languages)

Architecture of ASP.NET Core: use Model-View-Controller (MVC) architecture.

License of ASP.NET Core: Apache License 2.0

13. **SYMFONY:** A PHP framework that focuses on reusable components and follows the Model-View-Controller (MVC) architectural pattern.

Benefit of Symfony: i) Reusable components ii) scalability iii) extensive documentation.

Language of Symfony: The language used by Symfony is PHP

Architecture of Symfony: Symfony use Model-View-Controller (MVC) architecture

License of Symfony: MIT

14. **RUBY HANAMI (FORMERLY LOTUS):** A Ruby framework that emphasizes simplicity, modularity, and code organization.

Benefits of Ruby Hanami (formerly Lotus): i) Simplicity ii) modularity iii) ode organization.

Language of Ruby Hanami (formerly Lotus): The language used by Ruby Hanami (formerly Lotus**)** is Ruby.

Architecture of Ruby Hanami (formerly Lotus): Model-View-Controller (MVC) architecture

License of Ruby Hanami (formerly Lotus): MIT

15. **METEOR.JS:** A full-stack JavaScript framework that allows building real-time web applications.

Benefits of Meteor.js: i) Real-time applications ii) full-stack JavaScript iii) built-in synchronization.

Language of Meteor.js: The language used by Meteor.js is JavaScript

Architecture of Meteor.js: use Model-View-Controller (MVC) or Model-View-View Model (MVVM)

License of Meteor.js: MIT

16. **CAKE PHP:** A PHP framework that offers a rapid development environment with a clean and organized architecture. It follows the Model-View-Controller (MVC) pattern.

Benefits of Cake PHP: i) Rapid development, ii) clean architecture iii) large community.

Language of Cake PHP: The language used by Cake PHP is PHP

Architecture of Cake PHP: Model-View-Controller (MVC)

License of Cake PHP: MIT

17. **ASP.NET MVC:** A web application framework within the ASP.NET ecosystem, it provides a pattern-based way to build dynamic websites.

Benefits of ASP.NET MVC: i) Simplicity ii) speed iii) excellent performance.

Language of ASP.NET MVC: The language used by ASP.NET MVC is C#.

Architecture of ASP.NET MVC: Model-View-Controller (MVC) architecture.

License of ASP.NET MVC: Apache License 2.0

18. **PLAY FRAMEWORK:** A scalable and reactive web framework for Java and Scala.

Benefits of Play Framework: i) Scalability ii) reactive programming iii) support for Java and Scala.

Language of Play Framework: The language used by Play Framework Java, Scala

Architecture of Play Framework: Model-View-Controller (MVC) architecture

License of Play Framework: Apache License 2.0

19. **CODEIGNITER:** A lightweight PHP framework that focuses on simplicity and speed.

Benefits of Code Igniter: i) Simplicity ii) speed iii) excellent performance.

Language of Code Igniter: The language used by Code Igniter is PHP**.**

Architecture of Code Igniter: Model-View-Controller (MVC) architecture

License of Code Igniter: MIT

20. **PHOENIX:** A web framework for Elixir that emphasizes performance and real-time functionality.

Benefits of Phoenix: i) Performance ii) real-time functionality iii) fault-tolerant architecture.

Language of Phoenix: The language used by Phoenix is El ASP.NET MVC: C#.

Architecture of Phoenix: Model-View-Controller (MVC) architecture.

License of Phoenix: MIT

The best fit, whether for beginners or experts, varies depending on the framework.

- Beginner-friendly frameworks: Flask, Express.js, Ruby on Rails, Laravel, Code Igniter, Phoenix, Meteor.js. These frameworks offer simpler syntax, extensive documentation, and vibrant communities, making them more accessible to beginners.

- Intermediate to advanced frameworks: Django, Angular, React.js, Vue.js, Spring Boot, Ember.js, Symfony, ASP.NET, Play Framework, Ruby Hanami, Cake PHP. These frameworks require more experience with the underlying languages, but they provide more robust features, scalability, and flexibility for complex applications

Integrating a web framework with a web pro typically involves the following steps:

- Familiarize yourself with the framework's documentation and guides.

- Install the necessary dependencies and set up the development environment.

- Understand the framework's architecture and coding conventions.

- Define the project requirements and design the application structure accordingly.

- Implement the desired functionality using the framework's provided features and components.

- Test the application for performance, functionality, and security.

- Deploy the application to a web server or hosting platform.

- Continuously maintain and update the application as needed, following the framework's best practices and community guidelines.

Integration may also involve utilizing additional tools, libraries, or services depending on the framework and project requirements.

REFERENCES

Django: https://docs.djangoproject.com/

Rails: https://rubyonrails.org/documentation/

Laravel: https://laravel.com/docs

Express.js: https://expressjs.com/

Flask: https://flask.palletsprojects.com/

Angular: https://angular.io/docs

React.js: https://reactjs.org/docs/

Vue.js: https://vuejs.org/v2/guide/

Spring Boot: https://spring.io/projects/spring-boot

Ember.js: https://guides.emberjs.com/

ASP.NET: https://dotnet.microsoft.com/learn/aspnet

Symfony: https://symfony.com/doc/current/index.html

Meteor.js: https://docs.meteor.com/

CakePHP: https://book.cakephp.org/

Play Framework: https://www.playframework.com/documentation/

CodeIgniter: https://codeigniter.com/user_guide/

ASP.NET MVC: https://dotnet.microsoft.com/learn/aspnet-mvc

Phoenix: https://hexdocs.pm/phoenix/overview.html