Jason Dishongh - Web Exam

1.

PHP frameworks provide PHP platforms to assist in the development of web applications. PHP frameworks help save time by providing a basic starting structure rather than a web programmer having to build one from scratch and prevent the need for repetitive code across the web application [3]. PHP frameworks can provide pre-built assets or modules regarding repetitive tasks that a developer would have to create. This allows the developer to divert their time to building their application. Because of a framework's time saving, it promotes what is known as rapid application development (RAD) [1,2]. Reasons for selecting a framework differ between developers but a common opinion for choosing one is one that community support. Another key feature to look for is database support. Frameworks have the possibility of working with different databases and you might want to consider what works better with the company and what other developers have experience with [1]. For example, the Kohana framework doesn't support Oracle or SQlite, while the CodeIgniter supports Oracle and SQlite, and MySQL [1]. Documentation support is another feature to consider as a good framework should have a user guide that is readable and easy to understand and other relevant documentation should be updated frequently [1]. Another important feature for a framework is some form of a pre-built asset or module relating to testing regarding some form of test-driven development. One major factor that a good PHP framework should have is one that follows the Model View Controller (MVC) architectural fundamentals [1,2]. Model in MVC represents data with V relating to View or the User Interface while Controller relates to the actual logic known as the domain logic or the business logic [1,2]. In simple terms, MVC separates the logic from the user interface which allows one of the parts to be worked on separately from the other part. This allows development of one part without affecting the other part. This process is also known as the separation of corners and makes the development process easier [2]. Some important parts in the MVC process are routes and controllers. A route defines the handler information and URL pattern of the MVC application while the controller acts as the coordinator between the Model and the View and is also so responsible for controlling the application logic [4,5]. Controllers are also handling the flow of the application execution [4]. Routes can also hide information from the user with a common example of hiding file names from users [5].

2.

REST is known as Representational State Transfer which is an architectural style that provides standards or constraints between HTTP applications for use in web development [7]. There are 6 architectural constraints that REST outlines being stateless, layered system, client-server, uniform interface, cacheable and the code on demand constraint is considered as optional [6]. The stateless constraint essentially describes HTTP requests in which each request should be new meaning the server won't store anything so that there is no request history. A layered system is one which multiple servers are used to house certain aspects of development. For example, request authentication for Server A, API deployment for Server B and data storage on Server C [6]. Client-server describes a situation in which the client side and server side are separate and the evolution of each doesn't impact the other. A uniform interface is one in which the design of the interface is kept constant throughout the application

meaning across all available pages. Cacheable refers to the client-server interaction in which the caching of data and responses is used to improve client-side performance [6]. Code on demand is an optional constraint in which executable code that forms a widget for the UI is used when called by the API.

CRUD is known as create, read, update, and delete and most commonly refers to databases or similar forms of storage. The four functions are very straightforward with create describing creating a command, read as in reading/retrieving date, updating as in updating data, and finally delete as in deleting data [8,9]. CRUD often describes the functions or statements regarding databases. In the SQL language, the acronym is clearly defined in the CREATE, UPDATE, and DELETE statements with reading data using statements like SELECT. CRUD can also be described in web development in coordination with REST. In this instance, CRUD refers to the HTTP methods that are known as POST, GET, PUT, and DELETE which we have already used some of these methods in some our previous assignments. In the CRUD acronym, POST refers to Create, GET refers to read, PUT refers to Update, and DELETE refers to delete [8,9]. CRUD can be used in the user interface while REST relates to web development. The basics of REST were derived from the basic operations of CRUD [9].

Citations:

[1] https://onextrapixel.com/an-overview-of-php-framework-guides-for-developers/

[2] https://www.jotform.com/blog/discussing-php-frameworks/

[3] https://hackernoon.com/php-frameworks-explained-in-5-simple-questions-uvz31i7

[4] https://www.tutorialsteacher.com/mvc/routing-in-mvc

[5] https://www.tutorialspoint.com/mvc_framework/mvc_framework_controllers.htm

[6] https://restfulapi.net/rest-architectural-constraints/#:~:text=REST%20stands%20for%20Representational%20State,the%20development%20of%20web%20services.

[7] https://www.codecademy.com/articles/what-is-rest

[8] https://www.codecademy.com/articles/what-is-crud

[9] http://www.softwarecareertoday.net/faq/what-does-crud-mean-in-web-development/