**BIT 6103: LARAVEL EXAM QUESTION**

**Section A: Answer 5 out of 8 questions, question 1 is compulsory(20 marks).**

**other** questions carries **10 marks**.

1. Explain how to implement and use Repository and Service patterns in a Laravel application

2. Laravel is known for its expressive syntax and powerful tools. Discuss how Laravel's use of Service Providers contributes to the modularity and extensibility of the framework.

1. Explain the lifecycle of a Laravel request, starting from the incoming request to the response being sent to the client. Highlight the role of Middleware and Service Containers in this process.
2. Laravel allows for custom Blade directives. Explain how to create a custom Blade directive and discuss a use case where it would improve readability and reduce code duplication in views.

5 Laravel supports dynamic Blade components. Discuss how you would create a dynamic Blade component and how it can improve the handling of content in your views.

1. Laravel Eloquent ORM simplifies database interactions. Discuss how Eloquent handles relationships (one-to-one, one-to-many, and many-to-many) and demonstrate with an example how to define and retrieve a polymorphic relationship.

7. Define what an Accessor and a Mutator are in Laravel's Eloquent. Provide an example where you would use each and explain their impact on data handling within models.

8. Laravel’s Blade templating engine offers several advanced features. Discuss Blade components and slots, explaining how they enhance reusability and modularity in complex view structures.

**Section B: Answer 2 questions.** Each question carries 20 **marks**.

1. The `composer.json` file is crucial to any Laravel application. Discuss how you would manage dependencies for a Laravel project in a team environment, focusing on the roles of `composer.lock` and `autoload`.

2. Laravel's migration system is more than just creating tables. Explain how migrations manage schema versioning and rollbacks. Provide an example of handling a schema change that includes adding a new index and rolling it back.

3.Describe how to use Laravel’s Query Builder to perform complex database queries, including joins and aggregations. Discuss the advantages of using Query Builder over raw SQL in terms of security and performance.

4.Explain how to set up and utilize database transactions in Laravel, and provide an example where transactions would be necessary to ensure data integrity.