**MODULE – 9, 10(Forms, Controls)**

1. Explain ORM

Eloquent is an object relational mapper (ORM) that is included by default within the Laravel framework. An ORM is software that facilitates handling database records by representing data as objects, working as a layer of abstraction on top of the database engine used to store an application’s data.

Eloquent facilitates the task of interacting with database tables, providing an object-oriented approach to inserting, updating, and deleting database records, while also providing a streamlined interface for executing complex SQL queries.

1. Explain - Eloquent Relationships

Eloquent relationships are defined as methods on your Eloquent model classes. Since, like Eloquent models themselves, relationships also serve as powerful [query builders](https://laravel.com/docs/5.4/queries), defining relationships as methods provides powerful method chaining and querying capabilities. For example, we may chain additional constraints on this posts relationship:

1. What is Eager Loading and lazy loading?

**What is Lazy Loading**

Lazy loading is the practice of delaying load or initialization of resources or objects until they’re actually needed to improve performance and save system resources. For example, if a web page has an image that the user has to scroll down to see, you can display a placeholder and lazy load the full image only when the user arrives to its location.

The benefits of lazy loading include:

* **Reduces initial load time** – Lazy loading a webpage reduces page weight, allowing for a quicker [page load time](https://www.imperva.com/learn/performance/page-load-time/).
* **Bandwidth conservation** – Lazy loading conserves bandwidth by delivering content to users only if it’s requested.
* **System resource conservation** – Lazy loading conserves both server and client resources, because only some of the images, JavaScript and other code actually needs to be rendered or executed.

**Lazy Loading vs. Eager Loading**

While lazy loading delays the initialization of a resource, eager loading initializes or loads a resource as soon as the code is executed. Eager loading also involves pre-loading related entities referenced by a resource. For example, a PHP script with an include statement performs eager loading—as soon as it executes, eager loading pulls in and loads the included resources.

Eager loading is beneficial when there is an opportunity or need to load resources in the background. For example, some websites display a “loading” screen and eagerly load all the resources required for the web [application to run](https://www.imperva.com/learn/application-security/web-application-and-api-protection-waap/)