

## 1. Explain ORM

- The Object-Relational Mapping (ORM) system in Laravel, Eloquent ORM, provides a convenient and efficient way to manage your relational database using PHP. It is an integral part of the Laravel framework and offers many benefits, such as cleaner and more readable code, easier database migrations, and built-in support for common database operations.

## 2. Do Curd using Eloquent Query.

### Create

```
public function create()
{
    $data=country::all();
    return view('website/signup',['country'=>$data]);
}
```

### Insert

```
public function store(Request $request)
{
    $data=new customer;
        $data->name=$request->name;
        $data->email=$request->email;
        $data->username=$request->username;
        $data->password=Hash::make($request->password);
        $data->gen=$request->gen;
        $data->lag=explode(",",$request->lag);
        $data->mobile=$request->mobile;
        $data->cid=$request->cid;

        //img upload
        $file=$request->file('file');
        $filename=time().'_img.'.$request->file('file')->getClientOriginalExtension();
        ;
        $file->move('upload/customer',$filename); // use move for move image
        in public/images
        $data->file=$filename;

        $data->save();
        Alert::success('Congrats', 'You\'ve Successfully Registered');
        return redirect()->back();
}
```

```
}
```

## Update

```
function update_user(Request $req)
{
    $Users=User::find($req->uid); // get uid for where update from hidden input
    $unm=$Users->unm=$req->unm;
    $Users->email=$req->email;
    $Users->mobile=$req->mobile;
    $Users->save(); // update data

    return redirect('user');
    //return redirect()->back();
}
```

## Delete

```
public function destroy(customer $customer,$id)
{
    // get id data img
    $data=customer::find($id); // get only one data in string
    $filename=$data->file;

    // data delete with unlink image
    customer::find($id)->delete();
    if($filename!="")
    {
        unlink('upload/customer/'.$filename);
    }
    Alert::success('Congrats', 'You\'ve Successfully Deleted');
    return redirect()->back();
}
}
```

3. Explain - Eloquent Relationships.

Eloquent Relationships are an essential aspect of Laravel Eloquent. They enable you to define relationships between different tables in your database.

=>One-to-one Relationship

```
public function phone()
{
    return $this->hasOne('App\Phone');
}
```

=> One-to-many relationship

```
public function posts()
{
    return $this->hasMany('App\Post');
}
```

=> Many-to-Many Relationship

```
public function roles()
{
    return $this->belongsToMany('App\Role');
}
```

#### 4. What is Eager Loading and lazy loading?

- Eager Loading: Eager Loading is a technique used in Laravel Eloquent to reduce the number of queries made to the database when loading relationships. By using the 'with' method, you can specify which relationships to load when retrieving a model. This can be beneficial when dealing with deep relationships.

Example:

```
$users = App\User::with('posts')->get();
```

- Lazy Loading: is a technique used in Laravel Eloquent to load a relationship on demand. This means that the relationship data is only loaded from the database when you specifically request it.

```
$user = App\User::find(1);
$posts = $user->posts;
```