LARAVEL ASSIGNMENT

MODULE – 8 (MIGRATION)

Submitted to :-

Mr. Raj Nagar

Submitted by :-

Janvi Panchal

MODULE – 8(MIGRATION)

1. How to do config database in laravel

Ans:-

1. First, ensure you have a database set up and have the credentials (username, password, hostname, etc.) ready. Laravel supports several database systems, including MySQL, PostgreSQL, and SQLite.

2. Open the config/database.php file and find the connections array. This array contains all of the available connection configurations.

3. Choose a connection type from the array (e.g., mysql, sqlite, pgsql, etc.) and update the corresponding configuration options with your database credentials. For example, to configure a MySQL database, you might update the Default Database Connection Name:

```
'default' => env('DB_CONNECTION', 'mysql'),
```

4. Open the .env file in the root directory of your Laravel project. Set the DB_CONNECTION variable in this file to match the connection type you chose in the previous step. For example, if you are using a MySQL database, you would set the DB_CONNECTION variable to mysql.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

5. Set the DB_HOST variable to the hostname of your database server. For example, if you are using a local development server, you might set this to localhost.

6. Set the DB_DATABASE variable to the name of your database.

7. Set the DB_USERNAME and DB_PASSWORD variables to the username and password for your database.

8. If you are using a MySQL or PostgreSQL database, you may also need to set the DB_PORT variable in your .env file to the correct port number for your database server.

9. If you are using a MySQL database, you may also need to set the DB_SOCKET variable in your .env file if your MySQL server uses a socket other than the default.

10. Save the .env file and exit.

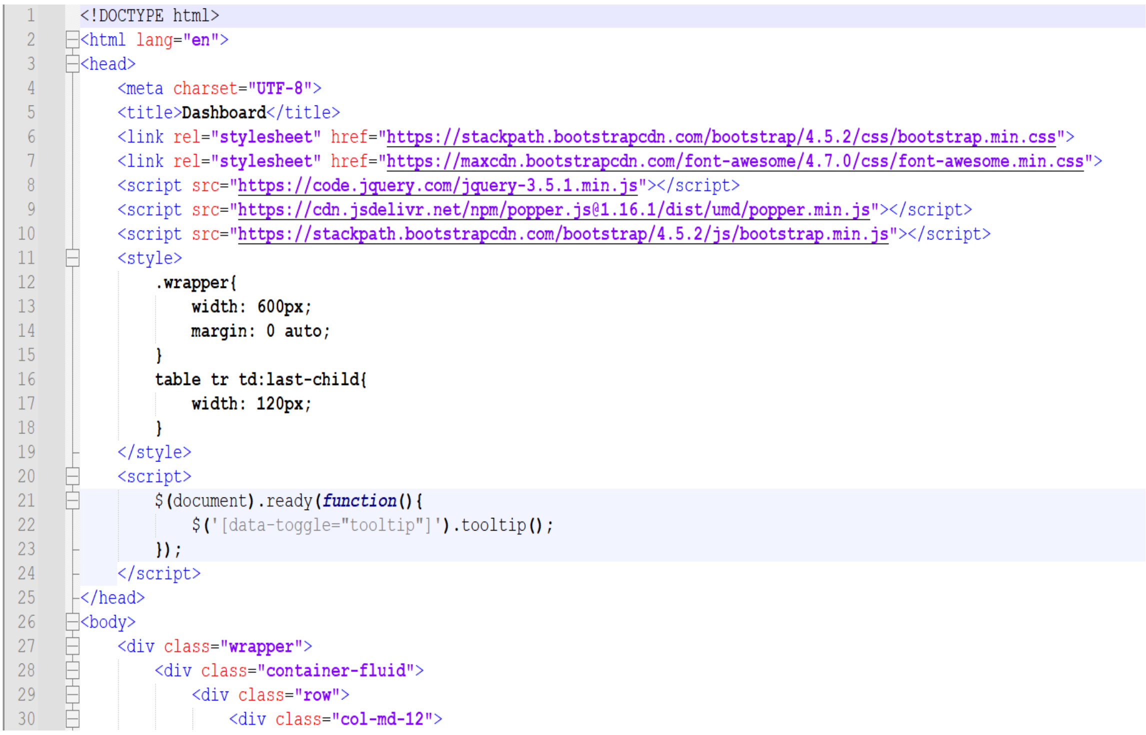
2.Call MySQLi Store Procedure from Laravel.

Ans:-

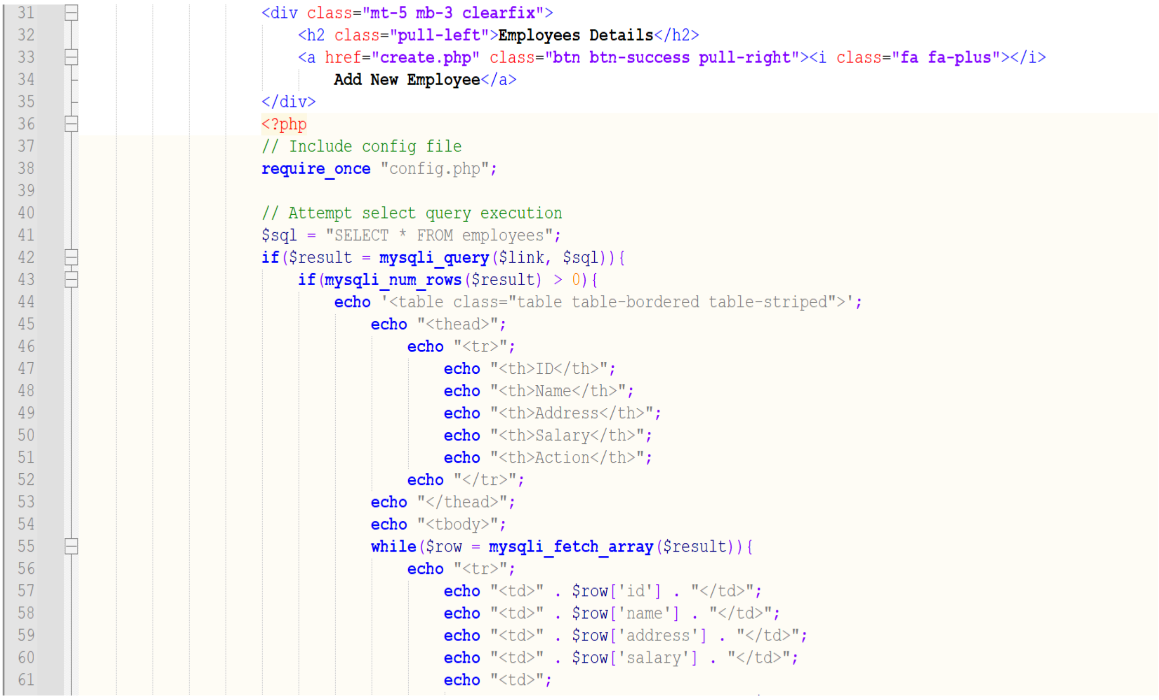
```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePostProcedure extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        $procedure = "DROP PROCEDURE IF EXISTS
`get_posts_by_userid`;
            CREATE PROCEDURE `get_posts_by_userid` (IN
idx int)
            BEGIN
            SELECT * FROM posts WHERE user_id = idx;
            END;";
        \DB::unprepared($procedure);
    }


    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {


    }
}
Route::get('call-procedure', function () {


    $postId = 1;
    $getPost = DB::select(
        'CALL get_posts_by_userid('.$postId.')'
    );


    dd($getPost);
});
```

3. Apply Curd Operation through Query Builder for Employee Management

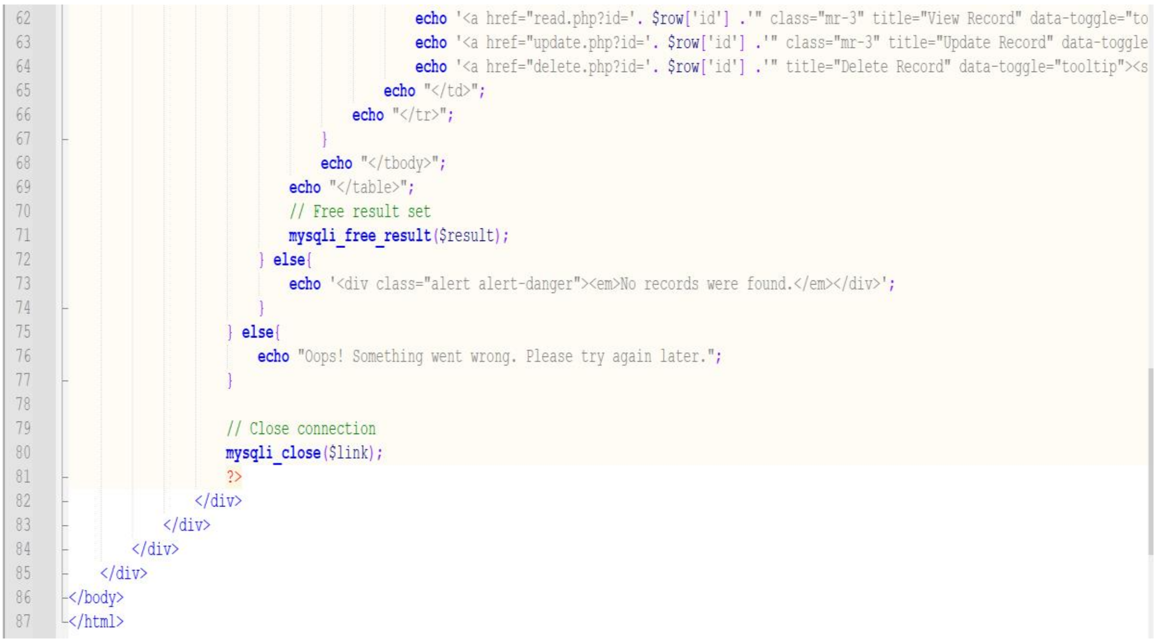
Ans:-



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    <style>
        .wrapper{
            width: 600px;
            margin: 0 auto;
        }
        table tr td:last-child{
            width: 120px;
        }
    </style>
    <script>
        $(document).ready(function(){
            $('[data-toggle="tooltip"]').tooltip();
        });
    </script>
</head>
<body>
    <div class="wrapper">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
```



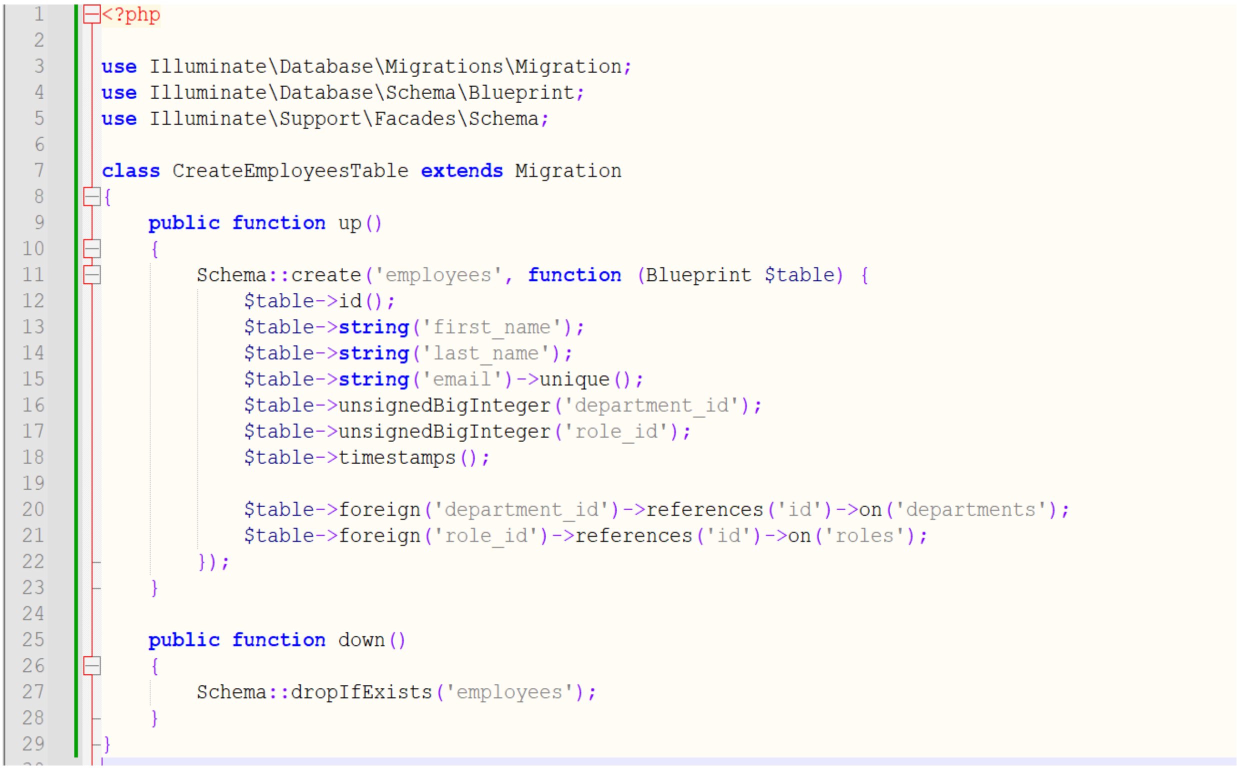
```
                    <div class="mt-5 mb-3 clearfix">
                        <h2 class="pull-left">Employees Details</h2>
                        <a href="create.php" class="btn btn-success pull-right"><i class="fa fa-plus"></i>
                            Add New Employee</a>
                    </div>
                    <?php
                    // Include config file
                    require_once "config.php";

                    // Attempt select query execution
                    $sql = "SELECT * FROM employees";
                    if($result = mysqli_query($link, $sql)){
                        if(mysqli_num_rows($result) > 0){
                            echo '<table class="table table-bordered table-striped">';
                                echo "<thead>";
                                    echo "<tr>";
                                        echo "<th>ID</th>";
                                        echo "<th>Name</th>";
                                        echo "<th>Address</th>";
                                        echo "<th>Salary</th>";
                                        echo "<th>Action</th>";
                                    echo "</tr>";
                                echo "</thead>";
                                echo "<tbody>";
                                while($row = mysqli_fetch_array($result)){
                                    echo "<tr>";
                                        echo "<td>" . $row['id'] . "</td>";
                                        echo "<td>" . $row['name'] . "</td>";
                                        echo "<td>" . $row['address'] . "</td>";
                                        echo "<td>" . $row['salary'] . "</td>";
                                        echo "<td>";
```



```
                                            echo '<a href="read.php?id='. $row['id'] .'" class="mr-3" title="View Record" data-toggle="to
                                            echo '<a href="update.php?id='. $row['id'] .'" class="mr-3" title="Update Record" data-toggle
                                            echo '<a href="delete.php?id='. $row['id'] .'" title="Delete Record" data-toggle="tooltip"><s
                                        echo "</td>";
                                    echo "</tr>";
                                }
                                echo "</tbody>";
                            echo "</table>";
                            // Free result set
                            mysqli_free_result($result);
                        } else{
                            echo '<div class="alert alert-danger"><em>No records were found.</em></div>';
                        }
                    } else{
                        echo "Oops! Something went wrong. Please try again later.";
                    }

                    // Close connection
                    mysqli_close($link);
                    ?>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

4. Create All Migration for Employee management

Ans:-



```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateEmployeesTable extends Migration
{
    public function up()
    {
        Schema::create('employees', function (Blueprint $table) {
            $table->id();
            $table->string('first_name');
            $table->string('last_name');
            $table->string('email')->unique();
            $table->unsignedBigInteger('department_id');
            $table->unsignedBigInteger('role_id');
            $table->timestamps();

            $table->foreign('department_id')->references('id')->on('departments');
            $table->foreign('role_id')->references('id')->on('roles');
        });
    }

    public function down()
    {
        Schema::dropIfExists('employees');
    }
}
```