Student Name: Ng Kerwin

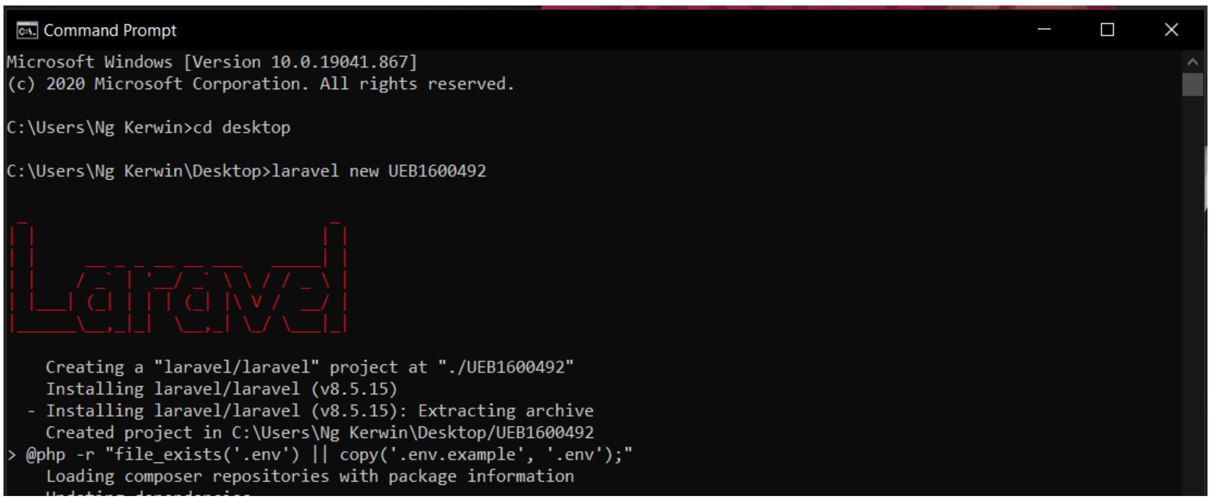
Student ID:1600492

Student practical: p4

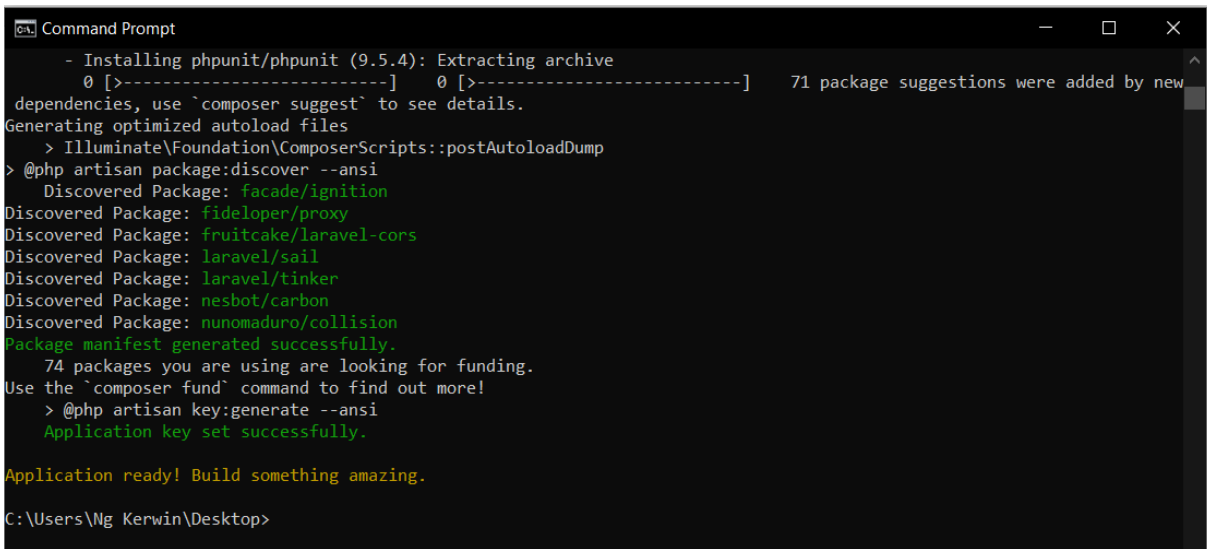
1)

A)

Project Initiation



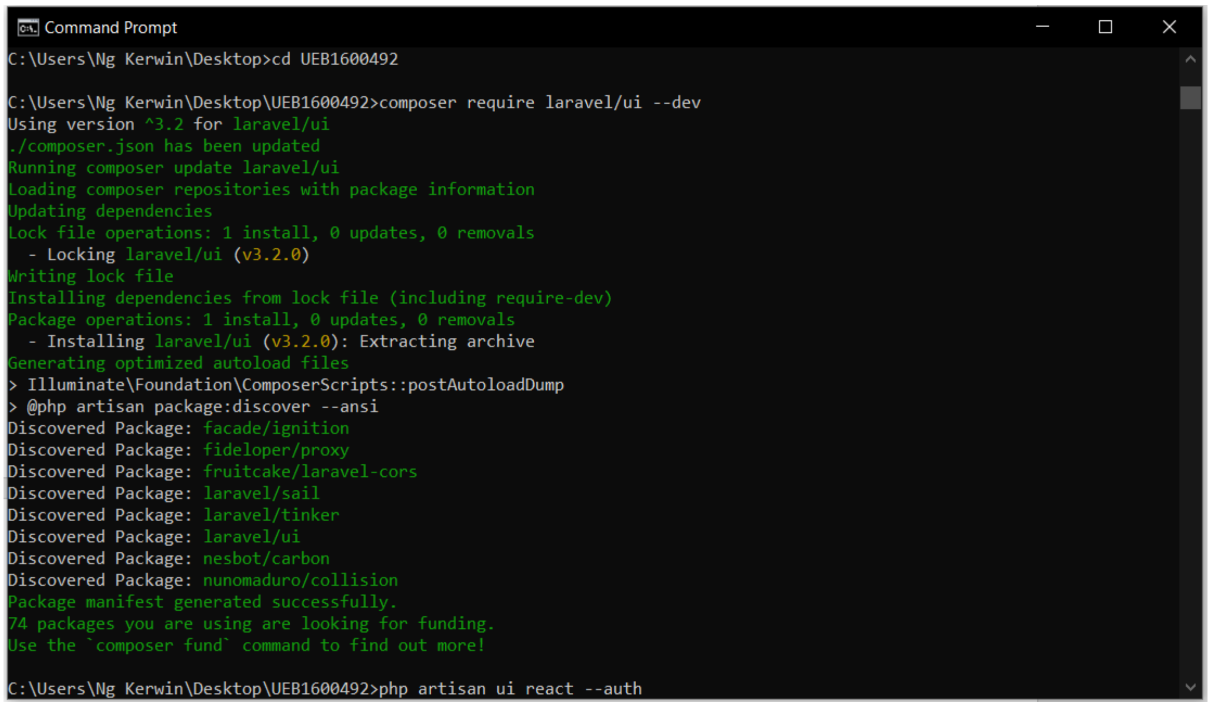
```
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Ng Kerwin>cd desktop

C:\Users\Ng Kerwin\Desktop>laravel new UEB1600492

 _                               _
| |                             | |
| |     __ _ _ __ __ ___   _____| |
| |    / _` | '__/ _` \ \ / / _ \ |
| |___| (_| | | | (_| |\ V /  __/ |
|______\__,_|_|  \__,_| \_/ \___|_|

    Creating a "laravel/laravel" project at "./UEB1600492"
    Installing laravel/laravel (v8.5.15)
  - Installing laravel/laravel (v8.5.15): Extracting archive
    Created project in C:\Users\Ng Kerwin\Desktop/UEB1600492
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
    Loading composer repositories with package information
```



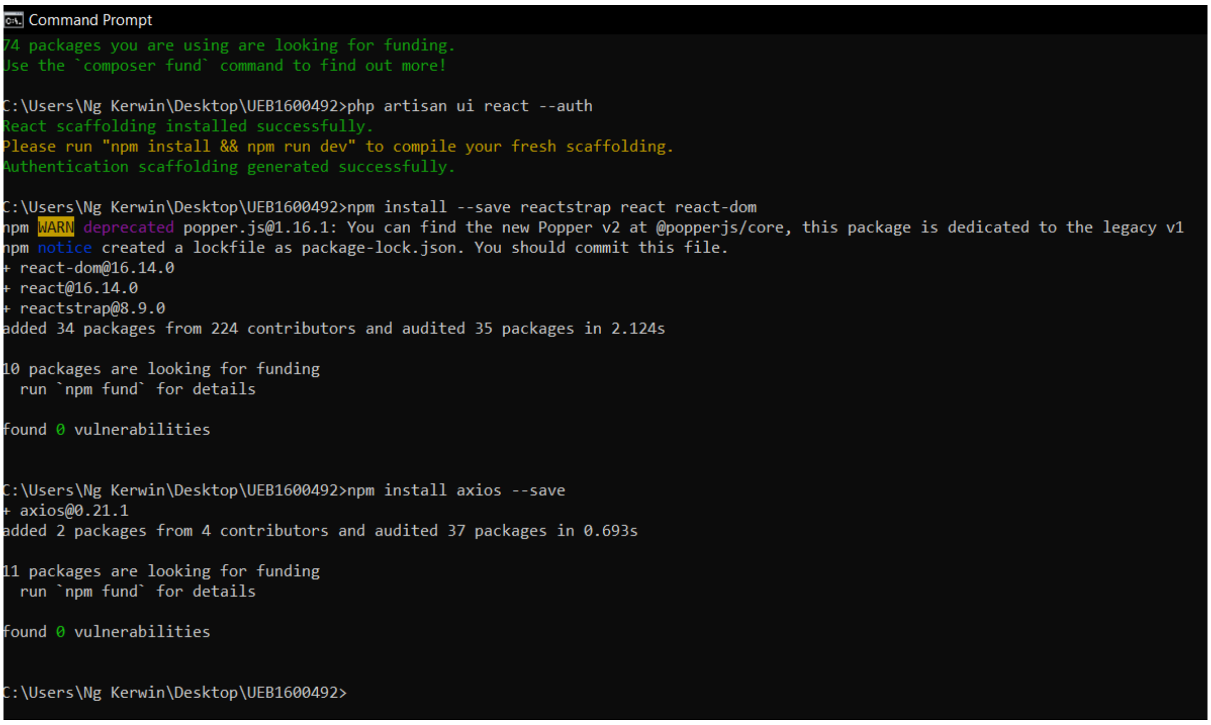
```
      - Installing phpunit/phpunit (9.5.4): Extracting archive
        0 [>---------------------------]    0 [>---------------------------]    71 package suggestions were added by new
 dependencies, use `composer suggest` to see details.
Generating optimized autoload files
      > Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
    Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
    74 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
    > @php artisan key:generate --ansi
    Application key set successfully.

Application ready! Build something amazing.

C:\Users\Ng Kerwin\Desktop>
```

B)

React Scaffolding



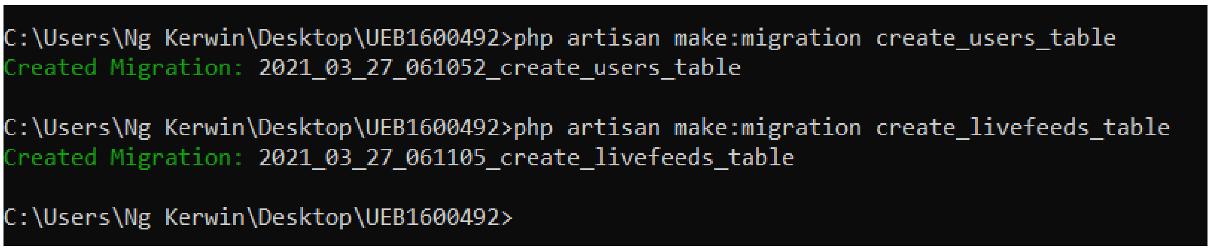
```
C:\Users\Ng Kerwin\Desktop>cd UEB1600492

C:\Users\Ng Kerwin\Desktop\UEB1600492>composer require laravel/ui --dev
Using version ^3.2 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v3.2.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravel/ui (v3.2.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
74 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan ui react --auth
```



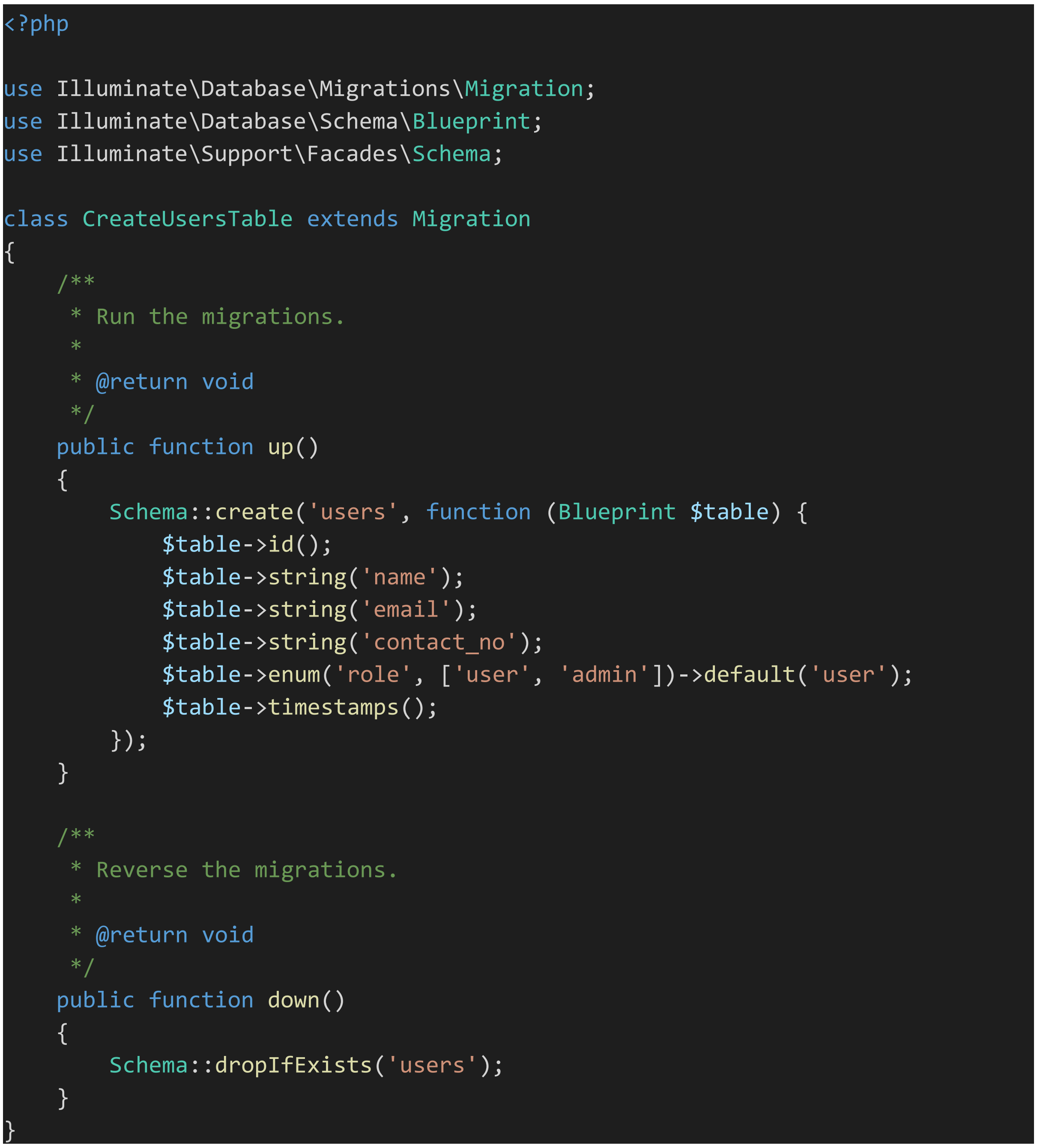
```
74 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan ui react --auth
React scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
Authentication scaffolding generated successfully.

C:\Users\Ng Kerwin\Desktop\UEB1600492>npm install --save reactstrap react react-dom
npm WARN deprecated popper.js@1.16.1: You can find the new Popper v2 at @popperjs/core, this package is dedicated to the legacy v1
npm notice created a lockfile as package-lock.json. You should commit this file.
+ react-dom@16.14.0
+ react@16.14.0
+ reactstrap@8.9.0
added 34 packages from 224 contributors and audited 35 packages in 2.124s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


C:\Users\Ng Kerwin\Desktop\UEB1600492>npm install axios --save
+ axios@0.21.1
added 2 packages from 4 contributors and audited 37 packages in 0.693s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


C:\Users\Ng Kerwin\Desktop\UEB1600492>
```

C)

Database Migration and its definition



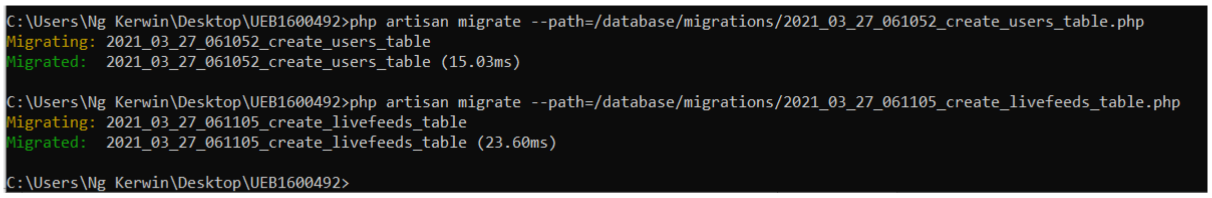
```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:migration create_users_table
Created Migration: 2021_03_27_061052_create_users_table

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:migration create_livefeeds_table
Created Migration: 2021_03_27_061105_create_livefeeds_table

C:\Users\Ng Kerwin\Desktop\UEB1600492>
```



```php
<?php

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

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->string('contact_no');
            $table->enum('role', ['user', 'admin'])->default('user');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

```php
<?php

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

class CreateLivefeedsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('livefeeds', function (Blueprint $table) {
            $table->id();
            $table->string('post');
            $table->string('description');
            $table->integer('edited');
            $table->integer('user_id')->foreign()->references('id')-
>on('users');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('livefeeds');
    }
}
```

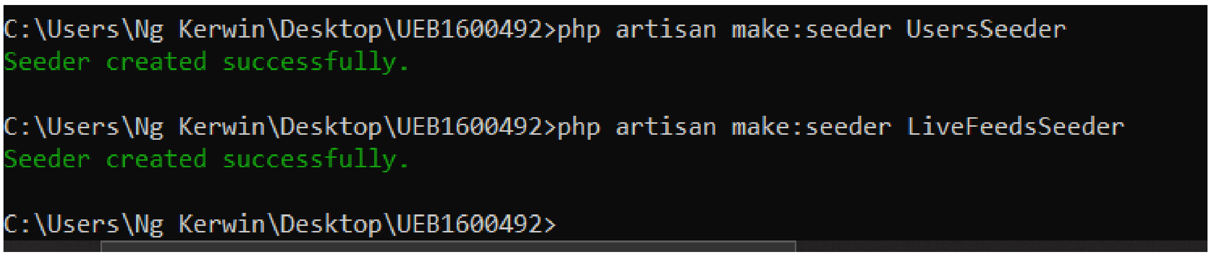
Migrating Migration Files



```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan migrate --path=/database/migrations/2021_03_27_061052_create_users_table.php
Migrating: 2021_03_27_061052_create_users_table
Migrated:  2021_03_27_061052_create_users_table (15.03ms)

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan migrate --path=/database/migrations/2021_03_27_061105_create_livefeeds_table.php
Migrating: 2021_03_27_061105_create_livefeeds_table
Migrated:  2021_03_27_061105_create_livefeeds_table (23.60ms)

C:\Users\Ng Kerwin\Desktop\UEB1600492>
```

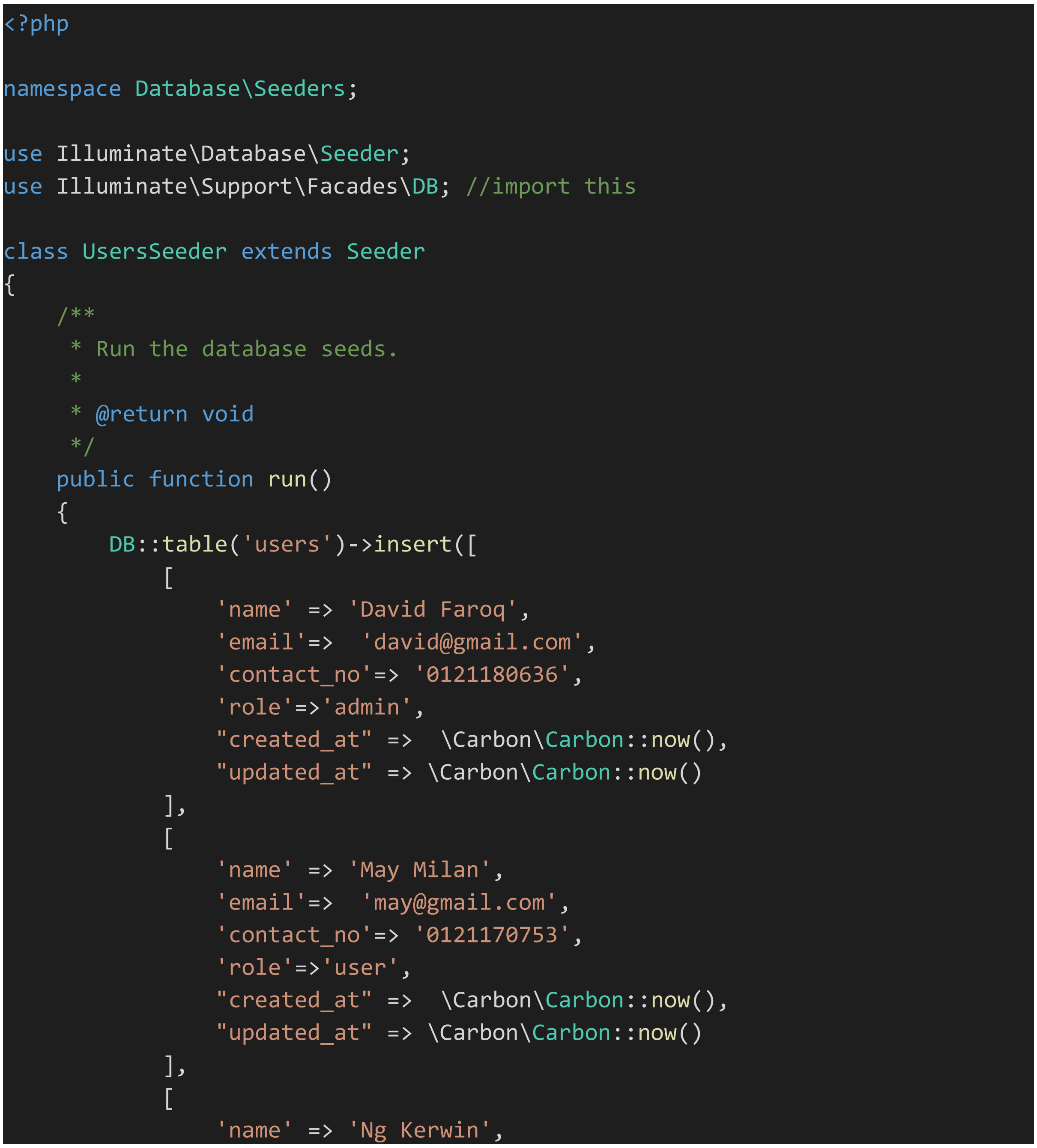
D)

Creating Seeders



```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:seeder UsersSeeder
Seeder created successfully.

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:seeder LiveFeedsSeeder
Seeder created successfully.

C:\Users\Ng Kerwin\Desktop\UEB1600492>
```

Definition of Seeders

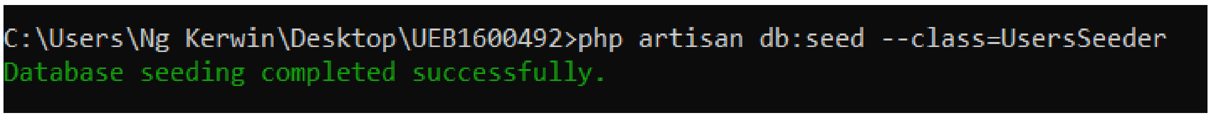


```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB; //import this

class UsersSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert([
            [
                'name' => 'David Faroq',
                'email'=>  'david@gmail.com',
                'contact_no'=> '0121180636',
                'role'=>'admin',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'name' => 'May Milan',
                'email'=>  'may@gmail.com',
                'contact_no'=> '0121170753',
                'role'=>'user',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'name' => 'Ng Kerwin',
                'email'=>  'kerwin@1utar.my',
                'contact_no'=> '0127209656',
                'role'=>'user',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'name' => 'Jia Jeng',
                'email'=>  'JiaJeng@1utar.my',
                'contact_no'=> '0126209656',
                'role'=>'user',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'name' => 'Tan Wei Yan',
                'email'=>  'weiyan@1utar.my',
                'contact_no'=> '0126209656',
                'role'=>'user',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],

        ]);
    }
}
```

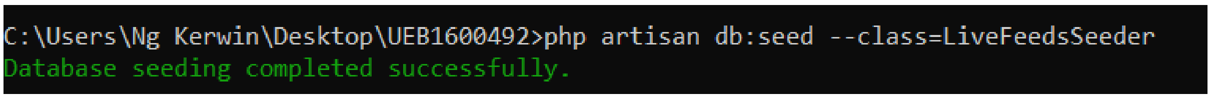
LiveFeed Seeders

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB; //import this

class LiveFeedsSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('livefeeds')->insert([
            [
                'post' => 'My bright Days',
                'description'=>  'This is the account of my brightest moments'
,
                'edited'=> '0',
                'user_id'=>'1',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'post' => 'Practical Exam',
                'description'=>  'This is the Practical Exam',
                'edited'=> '0',
                'user_id'=>'2',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'post' => 'Practical Test',
                'description'=>  'This is the practical Test',
                'edited'=> '0',
                'user_id'=>'3',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'post' => 'My bright Days in university',
                'description'=>  'This is the account of my brightest moments
in university',
                'edited'=> '0',
                'user_id'=>'4',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],
            [
                'post' => 'My bright Days in my high school',
                'description'=>  'This is the account of my brightest moments
in high school',
                'edited'=> '0',
                'user_id'=>'5',
                "created_at" =>  \Carbon\Carbon::now(),
                "updated_at" => \Carbon\Carbon::now()
            ],


        ]);
    }
}
```

Seeding the database



```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan db:seed --class=UsersSeeder
Database seeding completed successfully.
```

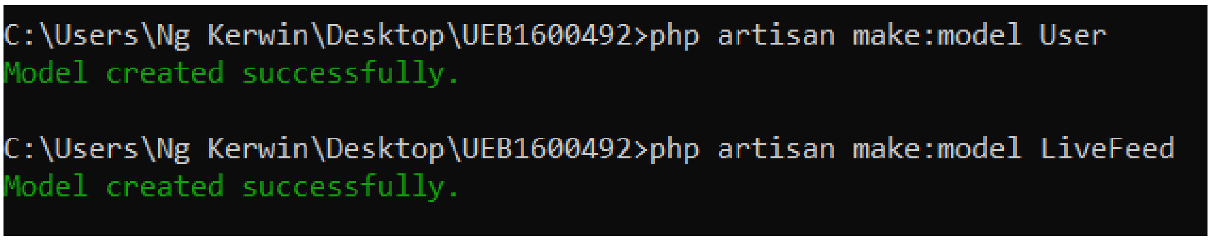


```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan db:seed --class=LiveFeedsSeeder
Database seeding completed successfully.
```

2)

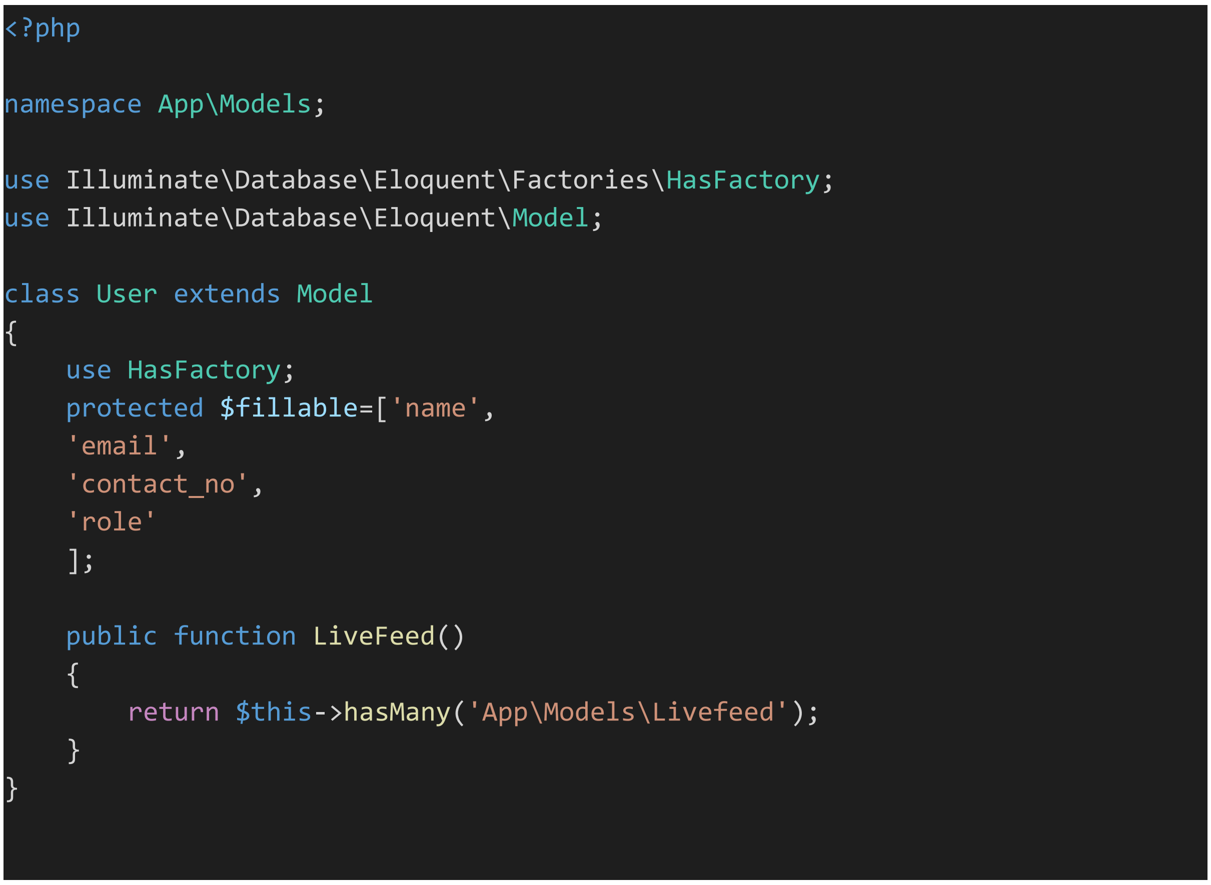
A)

Model



```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:model User
Model created successfully.

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:model LiveFeed
Model created successfully.
```

LiveFeed Model has been renamed to Livefeed to use eloquent relationship during coding

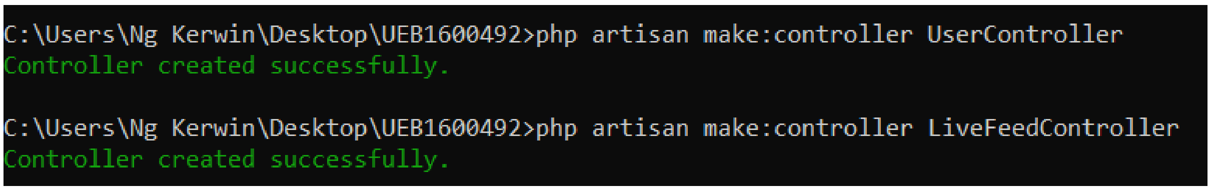


```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    use HasFactory;
    protected $fillable=['name',
    'email',
    'contact_no',
    'role'
    ];

    public function LiveFeed()
    {
        return $this->hasMany('App\Models\Livefeed');
    }
}
```

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Livefeed extends Model
{
    use HasFactory;
    protected $fillable=['post','description','edited','user_id'];

    public function User()
    {
        return $this->belongsTo('App\Models\User');
    }
}
```

B)

Controller Commands and Definition



```
C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:controller UserController
Controller created successfully.

C:\Users\Ng Kerwin\Desktop\UEB1600492>php artisan make:controller LiveFeedController
Controller created successfully.
```

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;

class UserController extends Controller
{
    ////Returns All Records
    public function index()
    {
       return User::all();
    }

    //Insert
    public function store(Request $request)
    {
        return User::create($request->all());
    }

    //Updates
    public function update(Request $request, $id)
    {
        $user =User::findOrFail($id);
        $user->update($request->all());
        return $user;
    }

    //Deletes
    public function destroy($id){
        $user=User::findOrFail($id);
        $user->delete();
        return 204;
    }
}
```

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Livefeed;

class LivefeedController extends Controller
{
    ////Returns All Records
    public function index()
    {
       return Livefeed::all();
    }

    //Insert
    public function store(Request $request)
    {
        return Livefeed::create($request->all());
    }

    //Updates
    public function update(Request $request, $id)
    {
        $livefeed =Livefeed::findOrFail($id);
        $livefeed->update($request->all());
        return $livefeed;
    }

    //Deletes
    public function destroy($id){
        $livefeed=Livefeed::findOrFail($id);
        $livefeed->delete();
        return 204;
    }
}
```

C)

Routes

```php
<?php

use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::view('/user', 'user');
Route::view('/livefeed','livefeed');
```

D)

View Files

app.blade.php

```
<!doctype html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }}</title>

    <!-- Scripts -->
    <script src="{{ asset('js/app.js') }}" defer></script>

    <!-- Fonts -->
    <link rel="dns-prefetch" href="//fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Nunito" re
l="stylesheet">

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}">
                    {{ config('app.name', 'Laravel') }}
                </a>
                <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="{{ __('Tog
gle navigation') }}">
                    <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" i
d="navbarSupportedContent">
                    <!-- Left Side Of Navbar -->
                    <ul class="navbar-nav mr-auto">
                    </ul>

                    <!-- Right Side Of Navbar -->
                    <ul class="navbar-nav ml-auto">
                       <li class="nav-item "><a class="nav-link active" hre
f="{{ url('/user') }}">User</a></li>
                       <li class="nav-item"><a class="nav-link active" hre
f="{{ url('/livefeed') }}">livefeed</a></li>
                    </ul>
                </div>
            </div>
        </nav>

        <main class="py-4">
            @yield('content')
        </main>
    </div>
</body>
<script src="/js/app.js"></script>
</html>
```

user.blade.php

```
@extends('layouts.app')

@section('content')
        <div class="relative flex items-top justify-center min-h-screen bg-
gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
            <div id="user"></div>
        </div>
@endsection
```

livefeed.blade.php

```
@extends('layouts.app')

@section('content')
        <div class="relative flex items-top justify-center min-h-screen bg-
gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
            <div id="livefeed"></div>
        </div>
@endsection
```

3)

A)
```php
<?php


use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;
use App\Http\Controllers\LiveFeedController;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('user',[UserController::class, 'index']);
Route::post('user',[UserController::class, 'store']);
Route::put('user/{id}',[UserController::class, 'update']);
Route::delete('user/{id}',[UserController::class, 'destroy']);

Route::get('livefeed',[LiveFeedController::class, 'index']);
Route::post('livefeed',[LiveFeedController::class, 'store']);
Route::put('livefeed/{id}',[LiveFeedController::class, 'update']);
Route::delete('livefeed/{id}',[LiveFeedController::class, 'destroy']);
```

B)

API endpoints for user.js

```
   loadUser() {
        axios.get('http://127.0.0.1:8000/api/user').then((response) => {
            this.setState({
                users: response.data
            })
        })
    }

    addUser() {
        axios.post('http://127.0.0.1:8000/api/user', thi
s.state.newUserData).then((response) => {
            let { users } = this.state
            this.loadUser()
            this.setState({
                users,
                newUserModal: false,
                newUserData: { name: "", email: "", contact_no: "", role: "" }
            })
        })
    }

    callUpdateUser(id, name, email, contact_no, role) {
        this.setState({
            updateUserModal: !this.state.updateUserModal,
            updateUserData: { id, name, email, contact_no, role }
        })
    }

    updateUser() {
        let { id, name, email, contact_no, role } = this.state.updateUserData
        axios.put('http://127.0.0.1:8000/api/user/' + thi
s.state.updateUserData.id, { name, email, contact_no, role }).then((response)
=> {
            this.loadUser()
            this.setState({
                updateUserModal: false,
                updateUserData: { id: "", name: "", email: "", contact_no: "",
role: "" }
            })
        })
    }

    deleteUser(id) {
        axios.delete('http://127.0.0.1:8000/api/user/' + id).then((response) =
> {
            this.loadUser();
        })
    }

    componentWillMount() {
        this.loadUser();
    }

    toggleNewUserModal() {
        this.setState({
            newUserModal: !this.state.newUserModal
        })
    }

    toggleUpdateUserModal() {
        this.setState({
            updateUserModal: !this.state.updateUserModal
        })
    }
```

API endpoints for livefeed.js

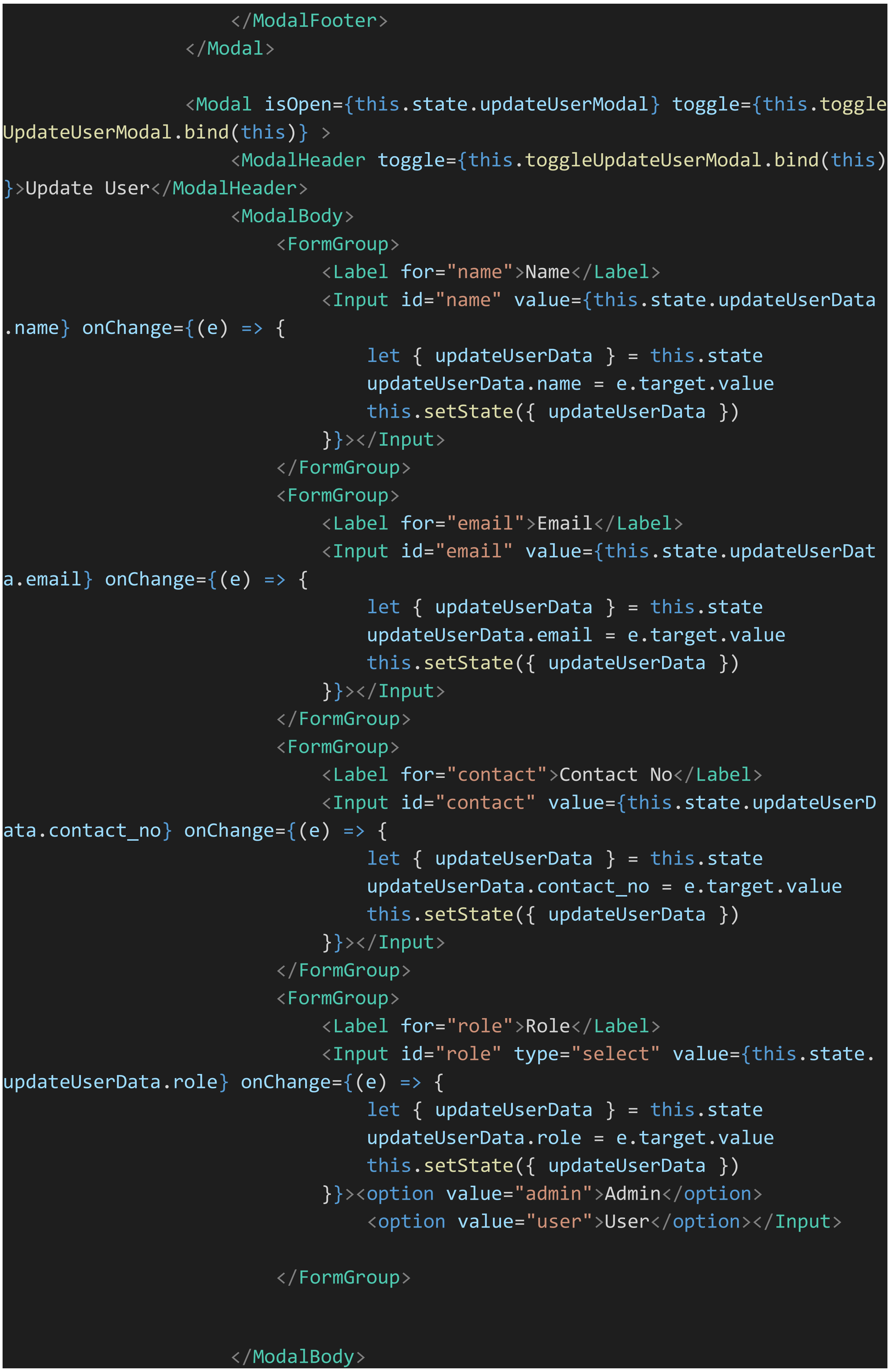
```
    loadLiveFeed(){
        axios.get('http://127.0.0.1:8000/api/livefeed').then((response) => {
            this.setState({
                livefeeds: response.data
            })
        })
    }

    addLiveFeed() {
        axios.post('http://127.0.0.1:8000/api/livefeed', thi
s.state.newLiveFeedData).then((response) => {
            let { livefeeds } = this.state
            this.loadLiveFeed()
            this.setState({
                livefeeds,
                newLiveFeedModal: false,
                newLiveFeedData: { post: "", description: "", edited: "", user
_id: "" }
            })
        })
    }

    callUpdateLiveFeed(id, post, description, edited, user_id) {
        this.setState({
            updateLiveFeedModal: !this.state.updateLiveFeedModal,
            updateLiveFeedData: { id, post, description, edited, user_id }
        })
    }

    updateLiveFeed() {
        let { id, post, description, edited, user_id } = this.state.updateLive
FeedData
        axios.put('http://127.0.0.1:8000/api/livefeed/' + thi
s.state.updateLiveFeedData.id, { post, description, edited:1, user_id}).then((
response) => {
            this.loadLiveFeed()
            this.setState({
                updateLiveFeedModal: false,
                updateLiveFeedData: {id:"",post:"", description: "", edited: "
", user_id: ""}
            })
        })
    }

    deleteLiveFeed(id) {
        axios.delete('http://127.0.0.1:8000/api/livefeed/'+id).then((response)
=> {
            this.loadLiveFeed();
        })
    }

    componentWillMount() {
        this.loadLiveFeed();
    }

    toggleNewLiveFeedModal() {
        this.setState({
            newLiveFeedModal: !this.state.newLiveFeedModal
        })
    }

    toggleUpdateLiveFeedModal() {
        this.setState({
            updateLiveFeedModal: !this.state.updateLiveFeedModal
        })
    }
```

C)

Design for user.js

```
    render() {
        let users = this.state.users.map((user) => {
            return (
                <tr key={user.id}>
                    <td>{user.id}</td>
                    <td>{user.name}</td>
                    <td>{user.email}</td>
                    <td>{user.contact_no}</td>
                    <td>{user.role}</td>
                    <td>
                        <Button
                            color="success"
                            size="sm"
                            className="mr-2"
                            onClick={this.callUpdateUser.bind(this, user.id,
user.name, user.email, user.contact_no, user.role)}>
                            Edit
                        </Button>
                        <Button
                            color="danger"
                            size="sm"
                            className="mr-2"
                            onClick={this.deleteUser.bind(this, user.id)}>
                            Delete
                        </Button>
                    </td>
                </tr>
            )
        })



        return (
            <div className="container">
                <div className="row">
                    <div className="col-md-8"><h3>User</h3></div>
                    <div className="col-md-offset-2"><Button color="primary" o
nClick={this.toggleNewUserModal.bind(this)}>Add User</Button></div>
                </div>
                <Modal isOpen={this.state.newUserModal} toggle={this.toggleNew
UserModal.bind(this)} >
```

```
                    <ModalHeader toggle={this.toggleNewUserModal.bind(this)}>Add New User</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="name">Name</Label>
                            <Input id="name" value={this.state.newUserData.name} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.name = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="email">Email</Label>
                            <Input id="email" value={this.state.newUserData.email} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.email = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="contact">Contact No</Label>
                            <Input id="contact" value={this.state.newUserData.contact_no} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.contact_no = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="role">Role</Label>
                            <Input id="role" type="select" value={this.state.newUserData.role} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.role = e.target.value
                                this.setState({ newUserData })
                            }}><option value="admin">Admin</option>
                                <option value="user">User</option></Input>

                        </FormGroup>

                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.addUser.bind(this)}>Add User</Button>{' '}
                        <Button color="secondary" onClick={this.toggleNewUserModal.bind(this)}>Cancel</Button>
```



```
                    </ModalFooter>
                </Modal>

                <Modal isOpen={this.state.updateUserModal} toggle={this.toggle
UpdateUserModal.bind(this)} >
                    <ModalHeader toggle={this.toggleUpdateUserModal.bind(this)
}>Update User</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="name">Name</Label>
                            <Input id="name" value={this.state.updateUserData
.name} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.name = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="email">Email</Label>
                            <Input id="email" value={this.state.updateUserDat
a.email} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.email = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="contact">Contact No</Label>
                            <Input id="contact" value={this.state.updateUserD
ata.contact_no} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.contact_no = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="role">Role</Label>
                            <Input id="role" type="select" value={this.state.
updateUserData.role} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.role = e.target.value
                                this.setState({ updateUserData })
                            }}><option value="admin">Admin</option>
                                <option value="user">User</option></Input>

                        </FormGroup>


                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.updateUser.bind
(this)}>Update User</Button>{' '}
                        <Button color="secondary" onClick={this.toggleUpdateUs
erModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Table>
                    <thead>
                        <tr>
                            <th>ID</th>
                            <th>Name</th>
                            <th>Email</th>
                            <th>Contact No</th>
                            <th>Role</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        {users}
                    </tbody>
                </Table>
            </div>
        );
    }
}

if (document.getElementById('user')) {
    ReactDOM.render(<User />, document.getElementById('user'));
}
```

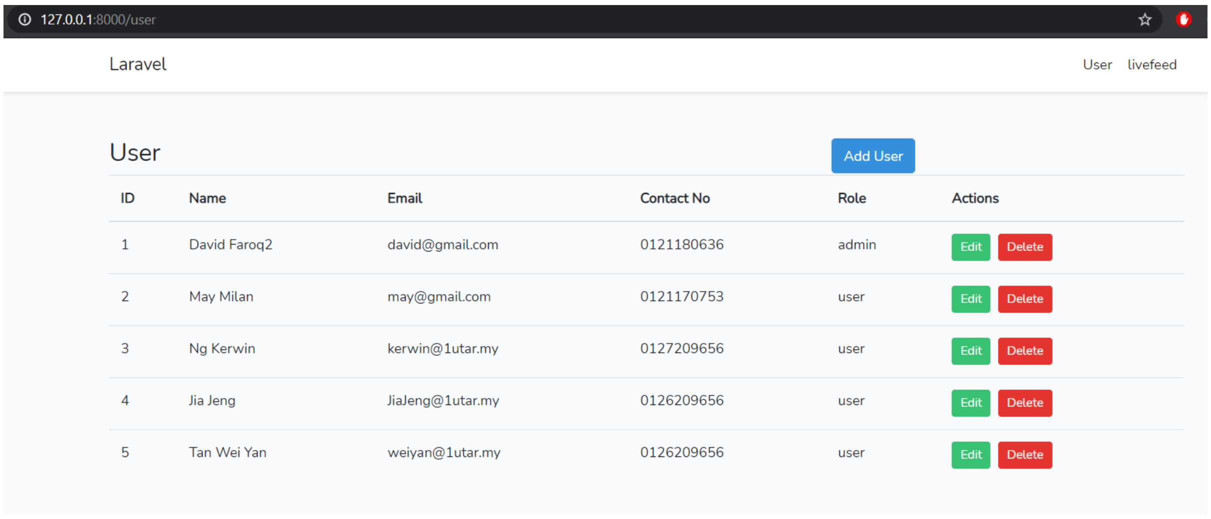
Design for livefeed.js

```
    render(){
        let livefeeds = this.state.livefeeds.map((livefeed) =>{
            return(
                <tr key={livefeed.id}>
                    <td>{livefeed.id}</td>
                    <td>{livefeed.post}</td>
                    <td>{livefeed.description}</td>
                    <td>{livefeed.edited}</td>
                    <td>{livefeed.user_id}</td>
                    <td>
                    <Button
                           color="success"
                           size="sm"
                           className="mr-2"
                           onClick={this.callUpdateLiveFeed.bind(this, livef
eed.id, livefeed.post, livefeed.description, livefeed.edited, livefeed.user_id
)}>
                           Edit
                       </Button>
                       <Button
                           color="danger"
                           size="sm"
                           className="mr-2"
                           onClick={this.deleteLiveFeed.bind(this, livefeed.
id)}>

                           Delete
                       </Button>
                    </td>
                </tr>
            )
        })



        return(
            <div className="container">
                <div className="row">
                   <div className="col-md-8"><h3>LiveFeed</h3></div>
                   <div className="col-md-offset-2"><Button color="primary" on
Click={this.toggleNewLiveFeedModal.bind(this)}>Add LiveFeed</Button></div>
                </div>
                <Modal isOpen={this.state.newLiveFeedModal} toggle={this.toggl
eNewLiveFeedModal.bind(this)} >
```

```
                    <ModalHeader toggle={this.toggleNewLiveFeedModal.bind(this
)}>Add New LiveFeed</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="post">Post</Label>
                            <Input id="post" value={this.state.newLiveFeedDat
a.post} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.post = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="description">Description</Label>
                            <Input id="description" value={this.state.newLive
FeedData.description} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.description = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>

                        <FormGroup>
                            <Label for="user_id">User ID</Label>
                            <Input id="user_id" value={this.state.newLiveFeed
Data.user_id} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.user_id = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>

                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.addLiveFeed.bin
d(this)}>Add LiveFeed</Button>{' '}
                        <Button color="secondary" onClick={this.toggleNewLiveF
eedModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Modal isOpen={this.state.updateLiveFeedModal} toggle={this.to
ggleUpdateLiveFeedModal.bind(this)} >
                    <ModalHeader toggle={this.toggleUpdateLiveFeedModal.bind(t
his)}>Update LiveFeed</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="post">Post</Label>
                            <Input id="post" value={this.state.updateLiveFeed
Data.post} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.post = e.target.value
                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="description">Description</Label>
                            <Input id="description" value={this.state.updateL
iveFeedData.description} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.description = e.target.val
ue

                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="user_id">User ID</Label>
                            <Input id="user_id" value={this.state.updateLiveF
eedData.user_id} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.user_id = e.target.value
                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>


                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.updateLiveFeed.
bind(this)}>Update LiveFeed</Button>{' '}
                        <Button color="secondary" onClick={this.toggleUpdateLi
veFeedModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Table>
                    <thead>
                        <tr>
                            <th>ID</th>
                            <th>Post</th>
                            <th>Description</th>
                            <th>Edited</th>
                            <th>User ID</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        {livefeeds}
                    </tbody>
                </Table>
            </div>
        );
    }
}

if (document.getElementById('livefeed')) {
    ReactDOM.render(<LifeFeed />, document.getElementById('livefeed'));
}
```

Screenshots

User List


127.0.0.1:8000/user

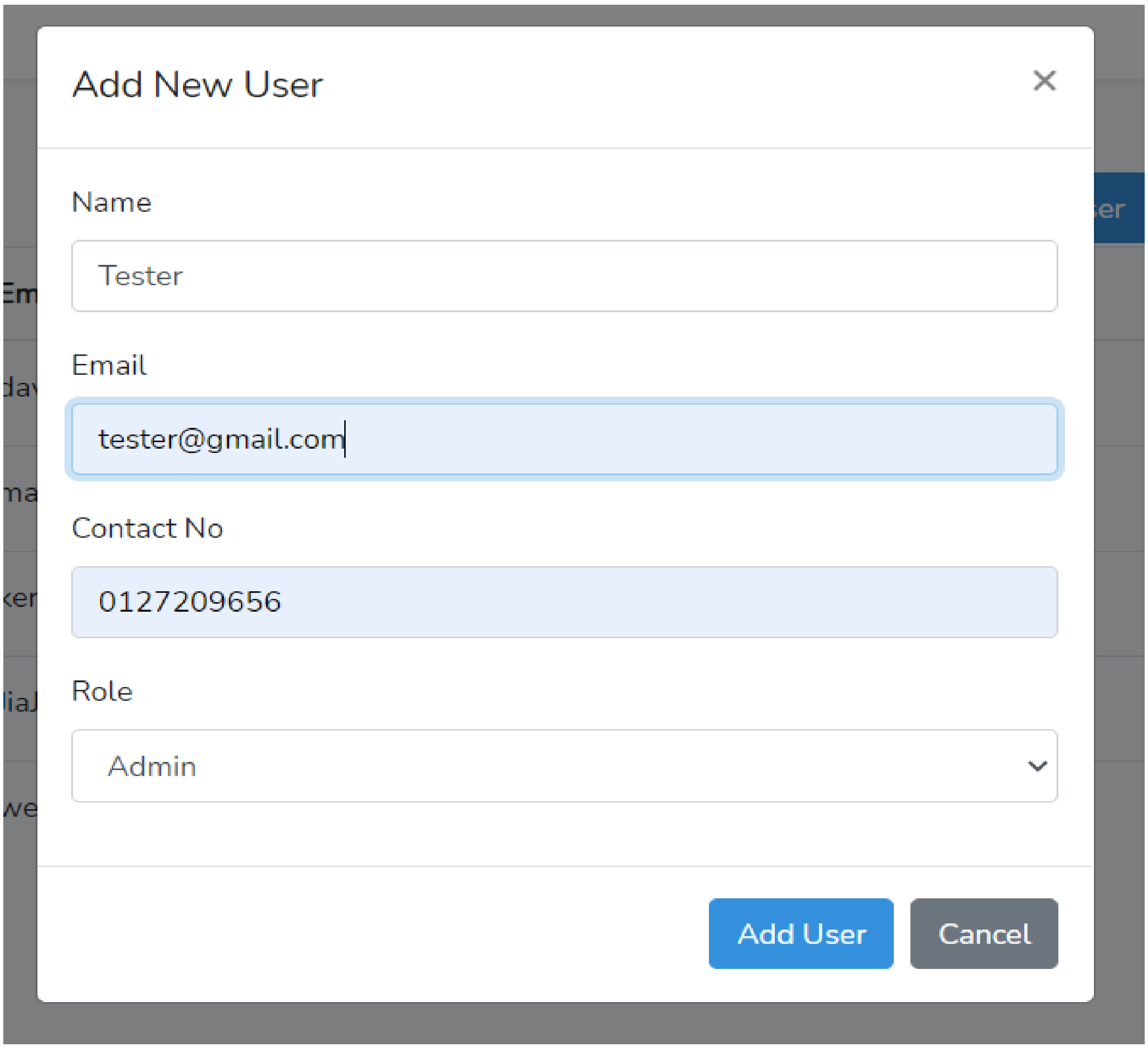
Laravel

User livefeed

User

Add User

ID	Name	Email	Contact No	Role	Actions
1	David Faroq2	david@gmail.com	0121180636	admin	Edit Delete
2	May Milan	may@gmail.com	0121170753	user	Edit Delete
3	Ng Kerwin	kerwin@1utar.my	0127209656	user	Edit Delete
4	Jia Jeng	JiaJeng@1utar.my	0126209656	user	Edit Delete
5	Tan Wei Yan	weiyan@1utar.my	0126209656	user	Edit Delete


Add New User

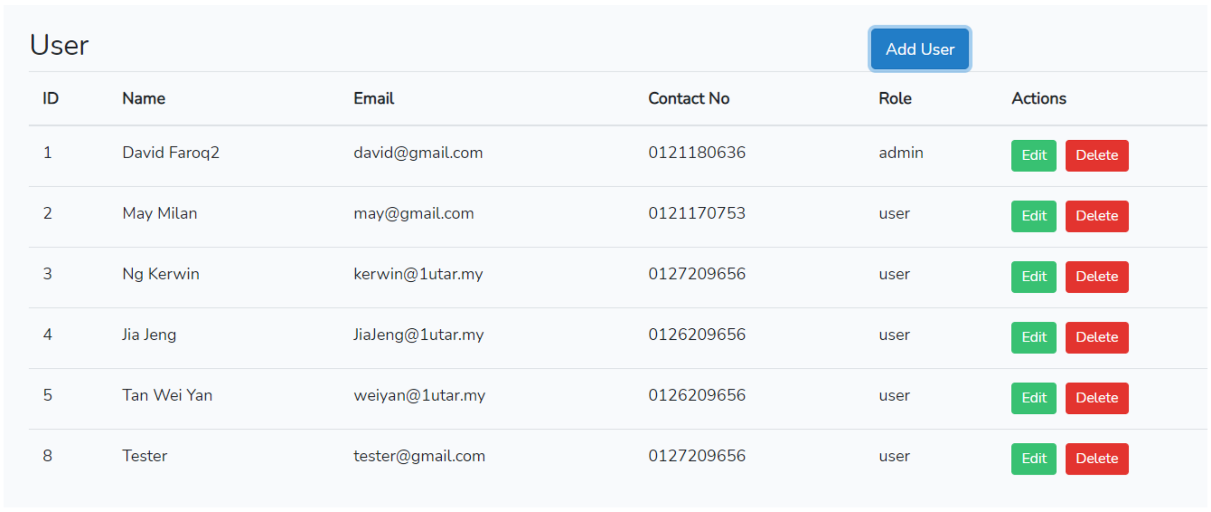
Name: Tester

Email: tester@gmail.com

Contact No: 0127209656

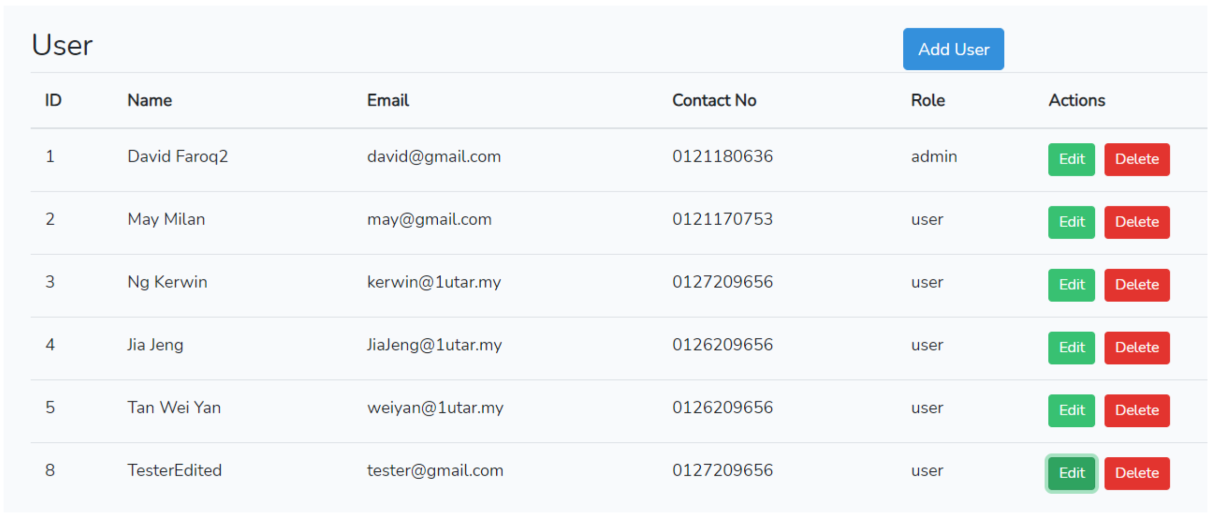
Role: Admin

Add User


User

ID	Name	Email	Contact No	Role	Actions
1	David Faroq2	david@gmail.com	0121180636	admin	Edit Delete
2	May Milan	may@gmail.com	0121170753	user	Edit Delete
3	Ng Kerwin	kerwin@1utar.my	0127209656	user	Edit Delete
4	Jia Jeng	JiaJeng@1utar.my	0126209656	user	Edit Delete
5	Tan Wei Yan	weiyan@1utar.my	0126209656	user	Edit Delete
8	Tester	tester@gmail.com	0127209656	user	Edit Delete

Added User

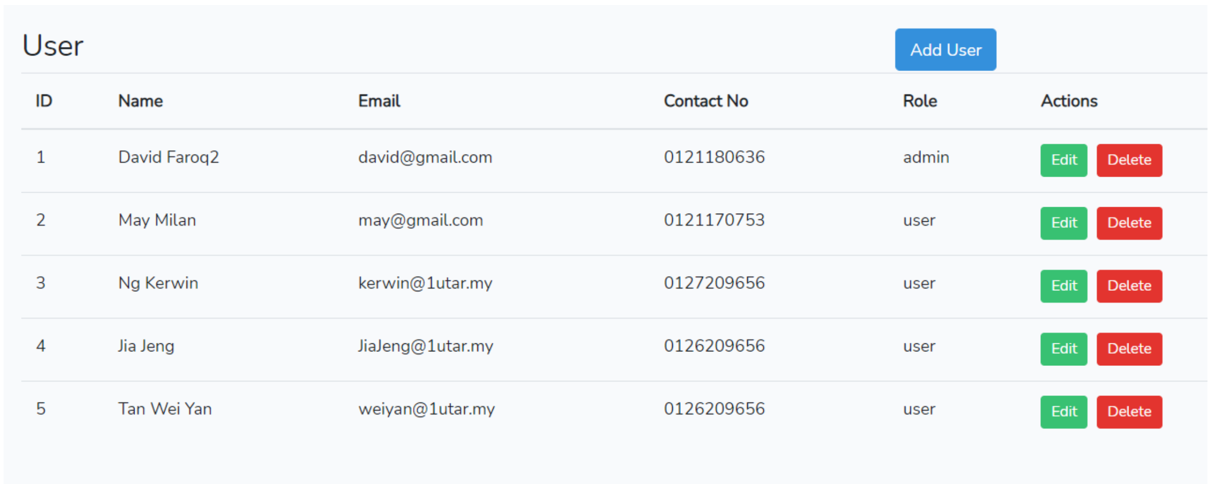


User

Add User

ID	Name	Email	Contact No	Role	Actions
1	David Faroq2	david@gmail.com	0121180636	admin	Edit Delete
2	May Milan	may@gmail.com	0121170753	user	Edit Delete
3	Ng Kerwin	kerwin@1utar.my	0127209656	user	Edit Delete
4	Jia Jeng	JiaJeng@1utar.my	0126209656	user	Edit Delete
5	Tan Wei Yan	weiyan@1utar.my	0126209656	user	Edit Delete
8	TesterEdited	tester@gmail.com	0127209656	user	Edit Delete

EditedUser

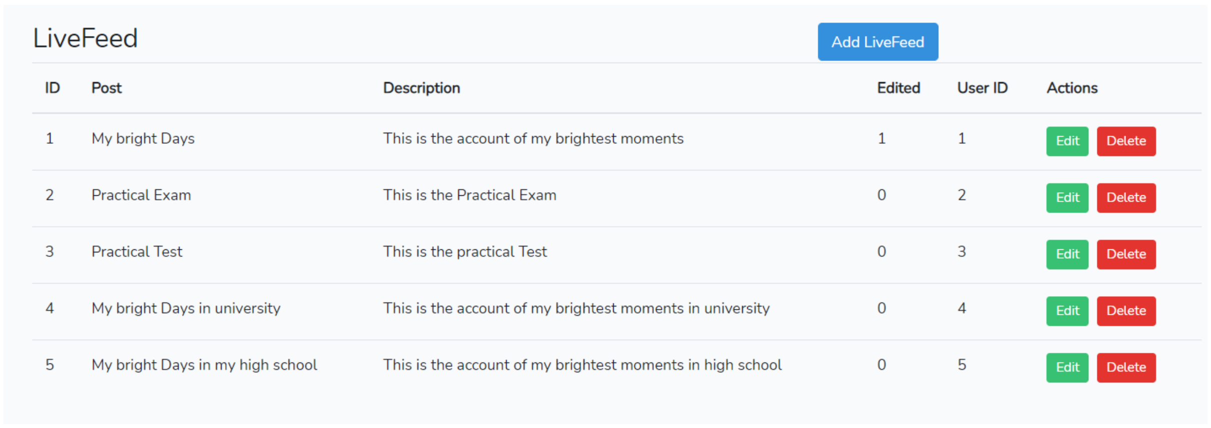


User

Add User

ID	Name	Email	Contact No	Role	Actions
1	David Faroq2	david@gmail.com	0121180636	admin	Edit Delete
2	May Milan	may@gmail.com	0121170753	user	Edit Delete
3	Ng Kerwin	kerwin@1utar.my	0127209656	user	Edit Delete
4	Jia Jeng	JiaJeng@1utar.my	0126209656	user	Edit Delete
5	Tan Wei Yan	weiyan@1utar.my	0126209656	user	Edit Delete

DeletedUser

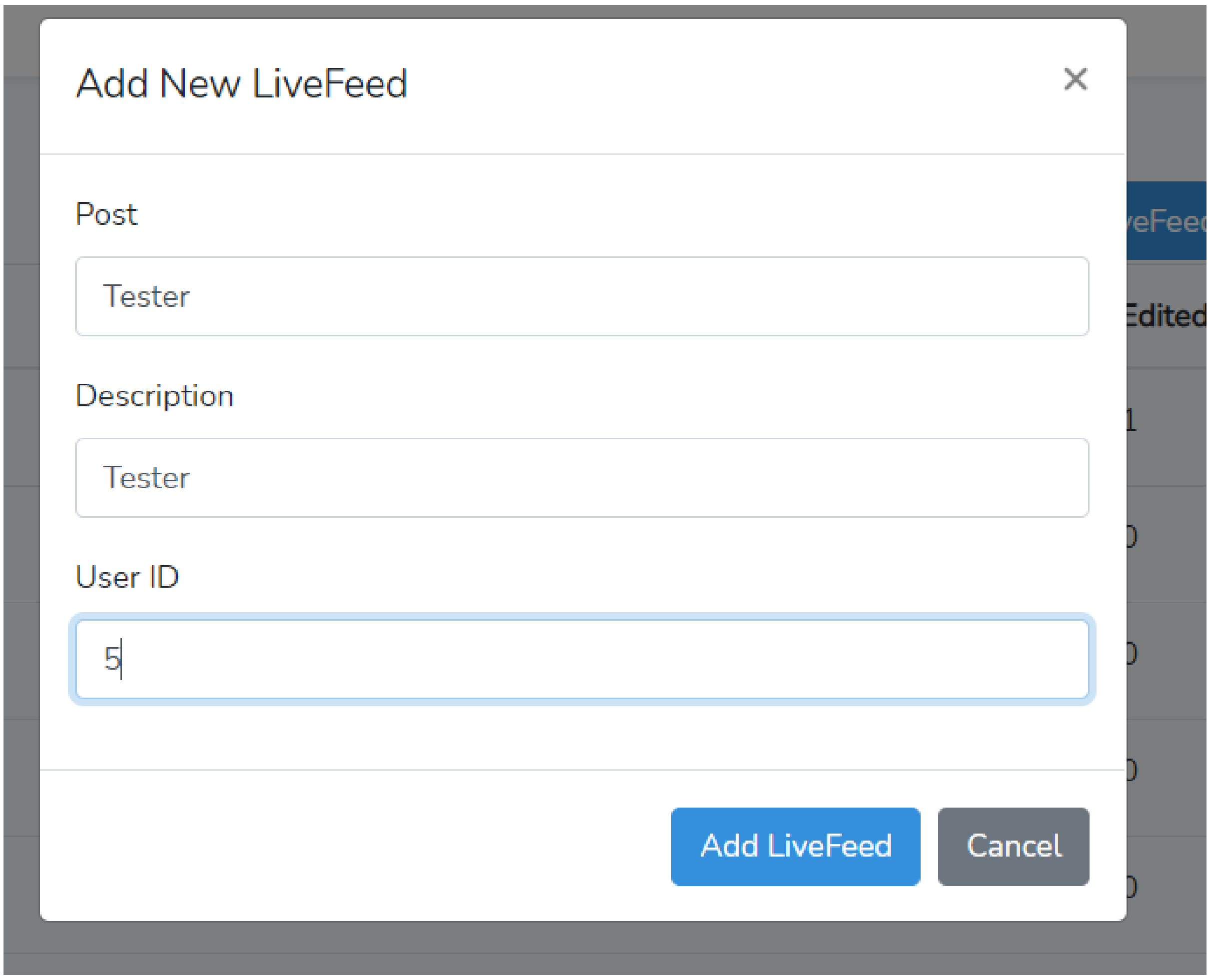


LiveFeed

Add LiveFeed

ID	Post	Description	Edited	User ID	Actions
1	My bright Days	This is the account of my brightest moments	1	1	Edit Delete
2	Practical Exam	This is the Practical Exam	0	2	Edit Delete
3	Practical Test	This is the practical Test	0	3	Edit Delete
4	My bright Days in university	This is the account of my brightest moments in university	0	4	Edit Delete
5	My bright Days in my high school	This is the account of my brightest moments in high school	0	5	Edit Delete

LiveFeedList

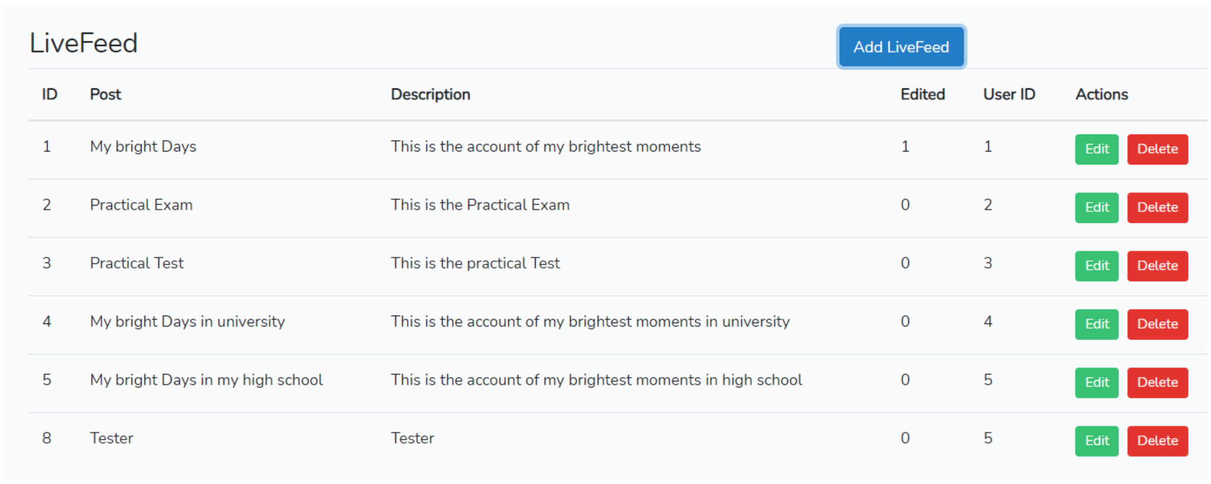


Add New LiveFeed

Post: Tester

Description: Tester

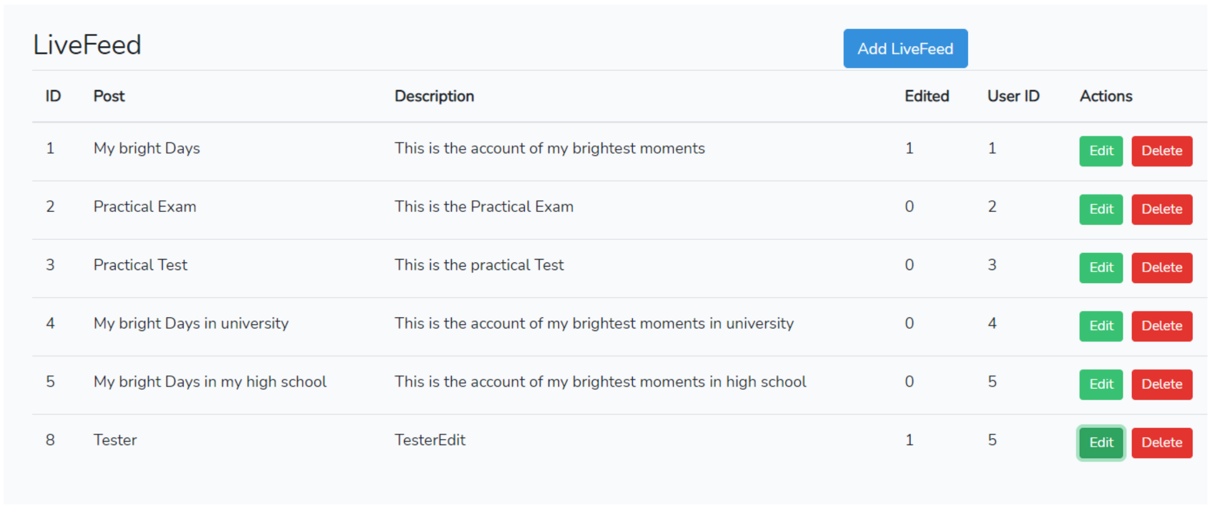
User ID: 5



LiveFeed

ID	Post	Description	Edited	User ID	Actions
1	My bright Days	This is the account of my brightest moments	1	1	Edit Delete
2	Practical Exam	This is the Practical Exam	0	2	Edit Delete
3	Practical Test	This is the practical Test	0	3	Edit Delete
4	My bright Days in university	This is the account of my brightest moments in university	0	4	Edit Delete
5	My bright Days in my high school	This is the account of my brightest moments in high school	0	5	Edit Delete
8	Tester	Tester	0	5	Edit Delete

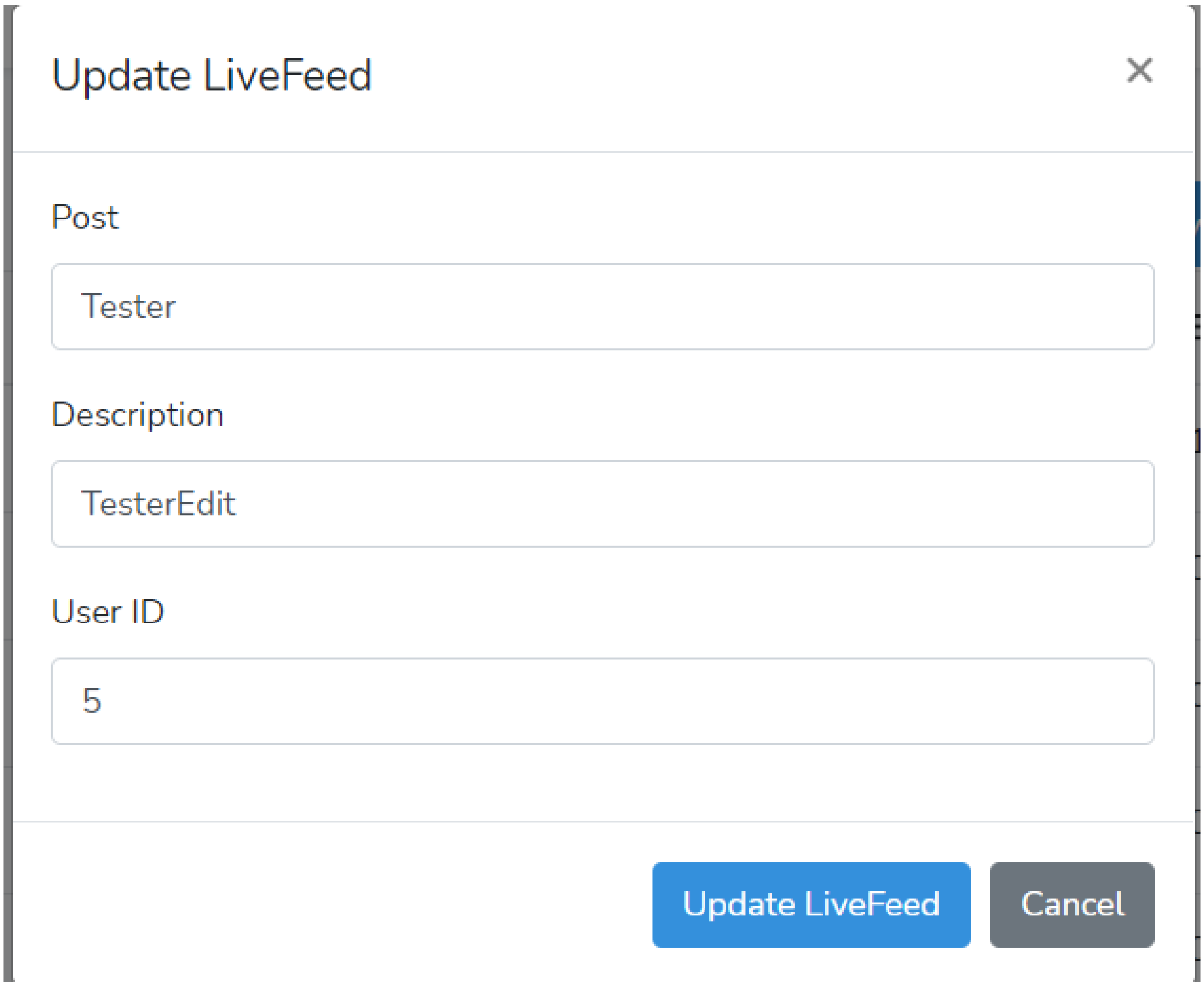
Added LiveFeed


LiveFeed

Add LiveFeed

ID	Post	Description	Edited	User ID	Actions
1	My bright Days	This is the account of my brightest moments	1	1	Edit Delete
2	Practical Exam	This is the Practical Exam	0	2	Edit Delete
3	Practical Test	This is the practical Test	0	3	Edit Delete
4	My bright Days in university	This is the account of my brightest moments in university	0	4	Edit Delete
5	My bright Days in my high school	This is the account of my brightest moments in high school	0	5	Edit Delete
8	Tester	TesterEdit	1	5	Edit Delete

Edited LiveFeed change edited to 1

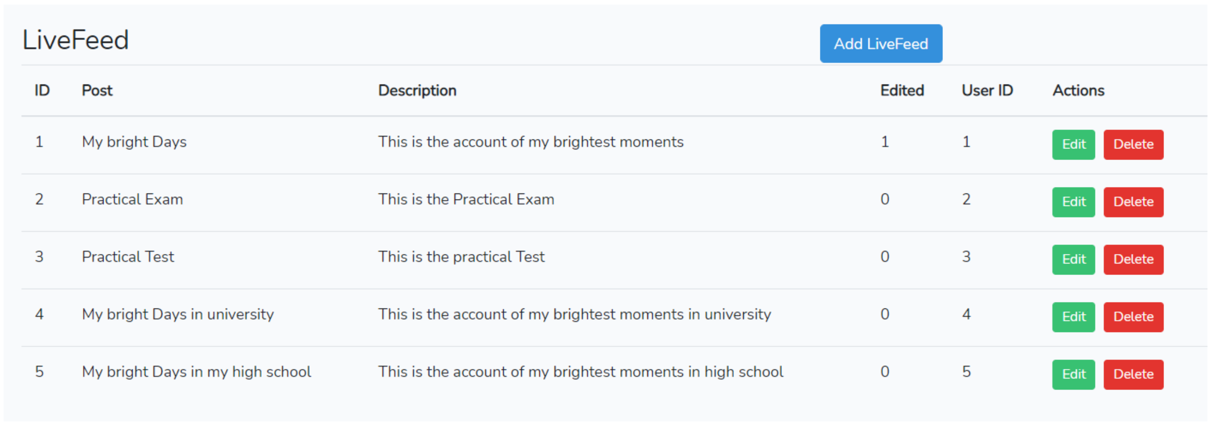

Update LiveFeed

Post: Tester

Description: TesterEdit

User ID: 5

Editing Page


LiveFeed

ID	Post	Description	Edited	User ID	Actions
1	My bright Days	This is the account of my brightest moments	1	1	Edit Delete
2	Practical Exam	This is the Practical Exam	0	2	Edit Delete
3	Practical Test	This is the practical Test	0	3	Edit Delete
4	My bright Days in university	This is the account of my brightest moments in university	0	4	Edit Delete
5	My bright Days in my high school	This is the account of my brightest moments in high school	0	5	Edit Delete

Deleted LiveFeed

FULL CODE FOR user.js

```
import axios from 'axios';
import React, { Component } from 'react';
import ReactDOM from 'react-dom';
import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, Input, For
mGroup, Label } from 'reactstrap';

export default class User extends Component {
    constructor() {
        super()
        this.state = {
            users: [],
            newUserModal: false,
            newUserData: { name: "", email: "", contact_no: "", role: "" },
            updateUserModal: false,
            updateUserData: { id: "", name: "", email: "", contact_no: "", rol
e: "" },
        }
    }

    loadUser() {
        axios.get('http://127.0.0.1:8000/api/user').then((response) => {
            this.setState({
                users: response.data
            })
        })
    }
    addUser() {
        axios.post('http://127.0.0.1:8000/api/user', thi
s.state.newUserData).then((response) => {
            let { users } = this.state
            this.loadUser()
            this.setState({
                users,
                newUserModal: false,
                newUserData: { name: "", email: "", contact_no: "", role: "" }
            })
        })
    }

    callUpdateUser(id, name, email, contact_no, role) {
        this.setState({
            updateUserModal: !this.state.updateUserModal,
            updateUserData: { id, name, email, contact_no, role }
        })
    }

    updateUser() {
        let { id, name, email, contact_no, role } = this.state.updateUserData
        axios.put('http://127.0.0.1:8000/api/user/' + thi
s.state.updateUserData.id, { name, email, contact_no, role }).then((response)
=> {
            this.loadUser()
            this.setState({
                updateUserModal: false,
                updateUserData: { id: "", name: "", email: "", contact_no: "",
role: "" }
            })
        })
    }

    deleteUser(id) {
        axios.delete('http://127.0.0.1:8000/api/user/' + id).then((response) =
> {
            this.loadUser();
        })
    }

    componentWillMount() {
        this.loadUser();
    }

    toggleNewUserModal() {
        this.setState({
            newUserModal: !this.state.newUserModal
        })
    }

    toggleUpdateUserModal() {
        this.setState({
            updateUserModal: !this.state.updateUserModal
        })
    }


    render() {
        let users = this.state.users.map((user) => {
            return (
                <tr key={user.id}>
                    <td>{user.id}</td>
                    <td>{user.name}</td>
                    <td>{user.email}</td>
                    <td>{user.contact_no}</td>
                    <td>{user.role}</td>
                    <td>
                        <Button
                            color="success"
                            size="sm"
                            className="mr-2"
                            onClick={this.callUpdateUser.bind(this, user.id,
user.name, user.email, user.contact_no, user.role)}>
                            Edit
                        </Button>
                        <Button
                            color="danger"
                            size="sm"
                            className="mr-2"
                            onClick={this.deleteUser.bind(this, user.id)}>
                            Delete
                        </Button>
                    </td>
                </tr>
            )
        })



        return (
            <div className="container">
                <div className="row">
                    <div className="col-md-8"><h3>User</h3></div>
                    <div className="col-md-offset-2"><Button color="primary" o
nClick={this.toggleNewUserModal.bind(this)}>Add User</Button></div>
```

```
                </div>
                <Modal isOpen={this.state.newUserModal} toggle={this.toggleNew
UserModal.bind(this)} >
                    <ModalHeader toggle={this.toggleNewUserModal.bind(this)}>A
dd New User</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="name">Name</Label>
                            <Input id="name" value={this.state.newUserData.na
me} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.name = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="email">Email</Label>
                            <Input id="email" value={this.state.newUserData.e
mail} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.email = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="contact">Contact No</Label>
                            <Input id="contact" value={this.state.newUserData
.contact_no} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.contact_no = e.target.value
                                this.setState({ newUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="role">Role</Label>
                            <Input id="role" type="select" value={this.state.
newUserData.role} onChange={(e) => {
                                let { newUserData } = this.state
                                newUserData.role = e.target.value
                                this.setState({ newUserData })
                            }}><option value="admin">Admin</option>
                                <option value="user">User</option></Input>

                        </FormGroup>

                    </ModalBody>
                    <ModalFooter>
```

```
                        <Button color="primary" onClick={this.addUser.bind(this)}>Add User</Button>{' '}
                        <Button color="secondary" onClick={this.toggleNewUserModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Modal isOpen={this.state.updateUserModal} toggle={this.toggleUpdateUserModal.bind(this)} >
                    <ModalHeader toggle={this.toggleUpdateUserModal.bind(this)}>Update User</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="name">Name</Label>
                            <Input id="name" value={this.state.updateUserData.name} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.name = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="email">Email</Label>
                            <Input id="email" value={this.state.updateUserData.email} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.email = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="contact">Contact No</Label>
                            <Input id="contact" value={this.state.updateUserData.contact_no} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.contact_no = e.target.value
                                this.setState({ updateUserData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="role">Role</Label>
                            <Input id="role" type="select" value={this.state.updateUserData.role} onChange={(e) => {
                                let { updateUserData } = this.state
                                updateUserData.role = e.target.value
                                this.setState({ updateUserData })
                            }}><option value="admin">Admin</option>
                                <option value="user">User</option></Input>
                        </FormGroup>


                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.updateUser.bind
(this)}>Update User</Button>{' '}
                        <Button color="secondary" onClick={this.toggleUpdateUs
erModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Table>
                    <thead>
                        <tr>
                            <th>ID</th>
                            <th>Name</th>
                            <th>Email</th>
                            <th>Contact No</th>
                            <th>Role</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        {users}
                    </tbody>
                </Table>
            </div>
        );
    }
}

if (document.getElementById('user')) {
    ReactDOM.render(<User />, document.getElementById('user'));
}
```

FULL CODE FOR livefeed.js

```
import axios from 'axios';
import React, { Component} from 'react';
import ReactDOM from 'react-dom';
import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, Input, For
mGroup, Label } from 'reactstrap';

export default class LifeFeed extends Component {
    constructor() {
        super()
        this.state = {
           livefeeds:[],
           newLiveFeedModal: false,
           newLiveFeedData: { post: "", description: "", edited: 0, user_id: "
" },
           updateLiveFeedModal:false,
           updateLiveFeedData: {id:"", post:"", description: "", edited: "", u
ser_id: ""},
        }
    }

    loadLiveFeed(){
        axios.get('http://127.0.0.1:8000/api/livefeed').then((response) => {
            this.setState({
                livefeeds: response.data
            })
        })
    }

    addLiveFeed() {
        axios.post('http://127.0.0.1:8000/api/livefeed', thi
s.state.newLiveFeedData).then((response) => {
            let { livefeeds } = this.state
            this.loadLiveFeed()
            this.setState({
                livefeeds,
                newLiveFeedModal: false,
                newLiveFeedData: { post: "", description: "", edited: "", user
_id: "" }
            })
        })
    }

    callUpdateLiveFeed(id, post, description, edited, user_id) {
        this.setState({
            updateLiveFeedModal: !this.state.updateLiveFeedModal,
            updateLiveFeedData: { id, post, description, edited, user_id }
        })
    }

    updateLiveFeed() {
        let { id, post, description, edited, user_id } = this.state.updateLive
FeedData
        axios.put('http://127.0.0.1:8000/api/livefeed/' + thi
s.state.updateLiveFeedData.id, { post, description, edited:1, user_id}).then((
response) => {
            this.loadLiveFeed()
            this.setState({
                updateLiveFeedModal: false,
                updateLiveFeedData: {id:"",post:"", description: "", edited: "
", user_id: ""}
            })
        })
    }

    deleteLiveFeed(id) {
        axios.delete('http://127.0.0.1:8000/api/livefeed/'+id).then((response)
=> {
            this.loadLiveFeed();
        })
    }

    componentWillMount() {
        this.loadLiveFeed();
    }

    toggleNewLiveFeedModal() {
        this.setState({
            newLiveFeedModal: !this.state.newLiveFeedModal
        })
    }

    toggleUpdateLiveFeedModal() {
        this.setState({
            updateLiveFeedModal: !this.state.updateLiveFeedModal
        })
    }


    render(){
        let livefeeds = this.state.livefeeds.map((livefeed) =>{
            return(
                <tr key={livefeed.id}>
                    <td>{livefeed.id}</td>
                    <td>{livefeed.post}</td>
                    <td>{livefeed.description}</td>
                    <td>{livefeed.edited}</td>
                    <td>{livefeed.user_id}</td>
                    <td>
                    <Button
                            color="success"
                            size="sm"
                            className="mr-2"
                            onClick={this.callUpdateLiveFeed.bind(this, livef
eed.id, livefeed.post, livefeed.description, livefeed.edited, livefeed.user_id
)}>
                            Edit
```

```
                        </Button>
                        <Button
                            color="danger"
                            size="sm"
                            className="mr-2"
                            onClick={this.deleteLiveFeed.bind(this, livefeed.
id)}>

                            Delete
                        </Button>
                    </td>
                </tr>
            )
        })



        return(
            <div className="container">
                <div className="row">
                   <div className="col-md-8"><h3>LiveFeed</h3></div>
                   <div className="col-md-offset-2"><Button color="primary" on
Click={this.toggleNewLiveFeedModal.bind(this)}>Add LiveFeed</Button></div>
                </div>
                <Modal isOpen={this.state.newLiveFeedModal} toggle={this.toggl
eNewLiveFeedModal.bind(this)} >
                    <ModalHeader toggle={this.toggleNewLiveFeedModal.bind(this
)}>Add New LiveFeed</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="post">Post</Label>
                            <Input id="post" value={this.state.newLiveFeedDat
a.post} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.post = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="description">Description</Label>
                            <Input id="description" value={this.state.newLive
FeedData.description} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.description = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>

                        <FormGroup>
```

```
                            <Label for="user_id">User ID</Label>
                            <Input id="user_id" value={this.state.newLiveFeedData.user_id} onChange={(e) => {
                                let { newLiveFeedData } = this.state
                                newLiveFeedData.user_id = e.target.value
                                this.setState({ newLiveFeedData })
                            }}></Input>
                        </FormGroup>

                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.addLiveFeed.bind(this)}>Add LiveFeed</Button>{' '}
                        <Button color="secondary" onClick={this.toggleNewLiveFeedModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Modal isOpen={this.state.updateLiveFeedModal} toggle={this.toggleUpdateLiveFeedModal.bind(this)} >
                    <ModalHeader toggle={this.toggleUpdateLiveFeedModal.bind(this)}>Update LiveFeed</ModalHeader>
                    <ModalBody>
                        <FormGroup>
                            <Label for="post">Post</Label>
                            <Input id="post" value={this.state.updateLiveFeedData.post} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.post = e.target.value
                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="description">Description</Label>
                            <Input id="description" value={this.state.updateLiveFeedData.description} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.description = e.target.value

                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>
                        <FormGroup>
                            <Label for="user_id">User ID</Label>
                            <Input id="user_id" value={this.state.updateLiveFeedData.user_id} onChange={(e) => {
                                let { updateLiveFeedData } = this.state
                                updateLiveFeedData.user_id = e.target.value
                                this.setState({ updateLiveFeedData })
                            }}></Input>
                        </FormGroup>


                    </ModalBody>
                    <ModalFooter>
                        <Button color="primary" onClick={this.updateLiveFeed.
bind(this)}>Update LiveFeed</Button>{' '}
                        <Button color="secondary" onClick={this.toggleUpdateLi
veFeedModal.bind(this)}>Cancel</Button>
                    </ModalFooter>
                </Modal>

                <Table>
                    <thead>
                        <tr>
                            <th>ID</th>
                            <th>Post</th>
                            <th>Description</th>
                            <th>Edited</th>
                            <th>User ID</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        {livefeeds}
                    </tbody>
                </Table>
            </div>
        );
    }
}

if (document.getElementById('livefeed')) {
    ReactDOM.render(<LifeFeed />, document.getElementById('livefeed'));
}
```