



UNIMAKER SYSTEM DOUMENTATION

PREPARED BY : PRIYANGKA JOHN JAYARAJ
DATE PREPARED : 8TH JANUARY 2020
SUPERVISED BY : MR.MUHAMMAD HAFIZAL BIN MUHD.HUSSIN

1.1 UniMaker System

UniMaker is a system that would be used through UniMaker 2019 Competition. UniMaker 2019 Competition is an initiative by Ministry of Higher Education and Jabatan Pendidikan Tinggi. This project was received in the 4th week and was expected to be completed by 9th week of industrial training.

1.2 Objectives

Main objectives of UniMaker system are as follows:

- i. Allow Admin to create, edit and delete course.
- ii. Allow Admin to upload, download and delete files.
- iii. Allow Admin to view, edit and delete registered team.
- iv. Allow Admin to view and delete registered users.
- v. Allow Users to enroll, unenrolled, view enroll and unrolled courses.

1.3 Project Implementation

UniMaker system has two user level which is admin and user. The system allows access to user only to few functionalities. The system is divided into four modules which is user, course, file and team in which each module consists of create, update, read and delete functionalities. Table below shows the code excerpt for each module.

a. Course Module

Table 1.0 Code Excerpt of Course Module

No	Feature	Code Excerpt
1	Create Course	<pre> public function createCourse() { return view('Course.create'); } public function store(Request \$request) { \$request->validate(['course_name' => 'required', 'desc' => 'required',]); \$course = new Course(['course_name' => \$request->get('course_name'), 'desc' => \$request->get('desc'),]); \$course->save(); return redirect('/file')->with('success', 'Course Saved!'); } </pre>
2	Edit Course	<pre> public function edit(\$id) { \$course = Course::find(\$id); return view('Course/edit', compact('course')); } /** * Update the specified resource in storage. * * @param \Illuminate\Http\Request \$request * @param int \$id * @return \Illuminate\Http\Response */ public function update(Request \$request, \$id) { \$request->validate(['course_name' => 'required', 'desc' => 'required',]); \$course = Course::find(\$id); \$course->course_name = \$request->get('course_name'); \$course->desc = \$request->get('desc'); \$course->save(); return redirect('/file')->with('success', 'Course updated!'); } </pre>

3	Delete Course	<pre> "/ public function destroy(\$id) { \$course = Course::find(\$id); \$course->delete(); return redirect('/file')->with('success', 'Course deleted!'); } </pre>
4	View Enrolled Course and Unenrolled Course	<pre> @foreach (\$listUnenroll as \$course) <div class="card bg-light"> <div class="card-body text-center"> <p class="card-text">{{ \$course->course_name }} </p> <p class="card-text">{{ \$course->desc }} </p> id }}'>Enroll id }}'>View Material </div> </div> @endforeach <h5>Course that you are enrolled in: </h5> @foreach (\$listEnroll as \$course) <div class="card bg-light"> <div class="card-body text-center"> <p class="card-text">{{ \$course->course_name }} </p> <p class="card-text">{{ \$course->desc }} </p> id }}'>Unenroll id }}'>View Material </div> </div> @endforeach </pre>
5	Enroll Course	<pre> public function enroll(Request \$request, \$id) { \$idUser = \Auth::user()->id; \$user = User::where('id', \$idUser)->first(); \$data = array("student_name" => \$user->name, "course_id" => \$id, "user_id" => \$user->id); DB::table('student')->insert(\$data); return redirect('/home')->with('success', 'Enrolled successfully!'); } </pre>
6	Unenrolled Course	<pre> public function unenroll(Request \$request, \$id) { \$userid = \Auth::user()->id; \$student = DB::table('student') ->where('student.user_id', '=', \$userid) ->where('student.course_id', '=', \$id) ->delete(); return redirect('/home')->with('success', 'Student unenrolled!'); } </pre>

b. File Module

Table 1.1 Code Excerpt of File Module

No	Feature	Code Excerpt
1	Upload File	<pre> public function create() { return view('file.upload'); } public function store(Request \$request, \$id) { \$course = Course::find(\$id); \$this->validate(\$request,['file'=>'required file max:20000']); \$upload= \$request->file('file'); \$path=\$upload->store('public/storage'); \$file=File::create(['title'=>\$upload->getClientOriginalName(), 'course_id'=>\$id, 'path'=>\$path]); return redirect('/file')->with('success','File is uploaded'); } </pre>
2	Download File	<pre> public function show(\$id) { \$dl=File::find(\$id); return Storage::download(\$dl->path,\$dl->title); } </pre>
3	Delete File	<pre> public function destroy(\$id) { \$del=File::find(\$id); Storage::delete(\$del->path); \$del->delete(); return redirect('/file')->with('success','File is deleted'); } </pre>

4	View File	<pre> public function index() { \$course = Course::all(); \$file=File::orderBy('created_at','desc')->paginate(30); return view('file.index',compact ('course','file')); } public function indexUser() { \$course = Course::all(); \$name= \Auth::user()->name ; \$data =DB::table('student') ->where('student_name', '=', '\$name') ->select('course_id','course_status') ->get(); \$file=File::orderBy('created_at','desc')->paginate(30); return view('file.indexUser',compact ('course','file','data')); } </pre>
---	-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

c. Team Module

Table 1.2 Code Excerpt of Team Module

No	Feature	Code Excerpt
1	Create Team	<pre> public function createTeam() { return view('registerteam'); } public function store(Request \$request) { \$request->validate(['name'=>'required', 'phone'=>'required', 'email'=>'required', 'uni'=>'required', 'team_name'=>'required', 'super_name'=>'required',]); \$team = new Team(['name'=>\$request->get('name'), 'phone'=>\$request->get('phone'), 'email'=>\$request->get('email'), 'uni'=>\$request->get('uni'), 'team_name'=>\$request->get('team_name'), 'super_name'=>\$request->get('super_name'),]); \$team->save(); return redirect('team.create')->with('success','Team Registered!'); } </pre>

2	Edit Team details	<pre> public function edit(\$id) { \$team = Team::find(\$id); return view('editteam', compact('team')); } /** * Update the specified resource in storage. * * @param \Illuminate\Http\Request \$request * @param int \$id * @return \Illuminate\Http\Response */ public function update(Request \$request, \$id) { \$request->validate(['name' => 'required', 'phone' => 'required', 'email' => 'required', 'uni' => 'required', 'team_name' => 'required', 'super_name' => 'required']); \$team = Team::find(\$id); \$team->name = \$request->get('name'); \$team->phone = \$request->get('phone'); \$team->email = \$request->get('email'); \$team->uni = \$request->get('uni'); \$team->team_name = \$request->get('team_name'); \$team->super_name = \$request->get('super_name'); \$team->save(); return redirect('team.view')->with('success', 'Changes is updated!'); } </pre>
3	Delete Team	<pre> public function destroy(\$id) { \$team = Team::find(\$id); \$team->delete(); return redirect('team.view')->with('success', 'Selected team details was deleted!'); } </pre>
4	View Team	<pre> public function index(Request \$request) { \$team = Team::all(); \$i=1; return view('viewteam',compact('team','i')); } public function show(\$id) { \$team = Team::find(\$id); return view('showteam', compact('team')); } </pre>

d. User Module

Table 1.3 Code Excerpt of User Module

No	Feature	Code Excerpt
1	Register User	<pre> protected \$redirectTo = '/home'; public function __construct() { \$this->middleware('guest'); } protected function validator(array \$data) { return Validator::make(\$data, ['name' => ['required', 'string', 'max:255'], 'email' => ['required', 'string', 'email', 'max:255', 'unique:users'], 'password' => ['required', 'string', 'min:8', 'confirmed'],]); } protected function create(array \$data) { return User::create(['name' => \$data['name'], 'email' => \$data['email'], 'password' => Hash::make(\$data['password']),]); } public function index() { \$users = User::all(); return view('viewuser', compact('users')); } </pre>
2	View User Details	<pre> public function showUsers() { \$users = User::all(); \$i=1; return view('viewuser', compact('users','i')); } public function displayUser(\$id) { \$users = User::find(\$id); return view('showuser', compact('users')); } </pre>

3	Delete User	<pre> public function destroyUser(\$id) { \$users = User::find(\$id); \$users->delete(); return redirect('user.view')->with('success', 'User deleted!'); } </pre>
4	Migration Create User Table	<pre> class CreateUsersTable extends Migration { /** * Run the migrations. * * @return void */ public function up() { Schema::create('users', function (Blueprint \$table) { \$table->bigIncrements('id'); \$table->string('name'); \$table->string('email')->unique(); \$table->string('role')->nullable(); \$table->timestamp('email_verified_at')->nullable(); \$table->string('password'); \$table->rememberToken(); \$table->timestamps(); }); } /** * Reverse the migrations. * * @return void */ public function down() { Schema::dropIfExists('users'); } } </pre>

1.4 UniMaker System Interface

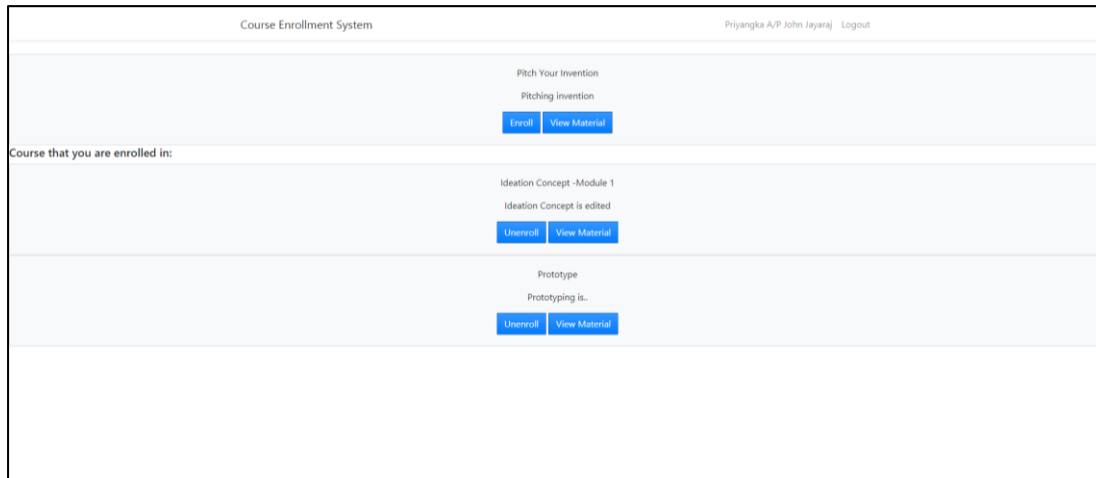


Figure 1.1 User Dashboard

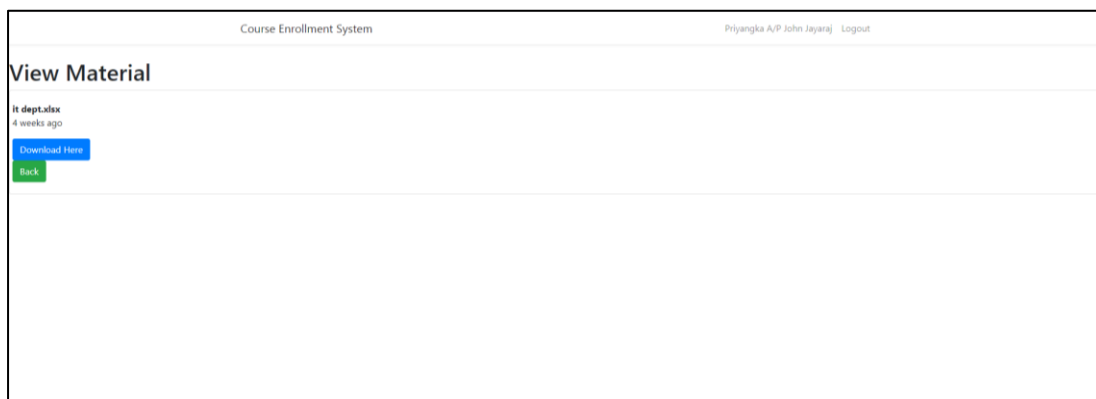


Figure 1.2 View Material Functionality (User)

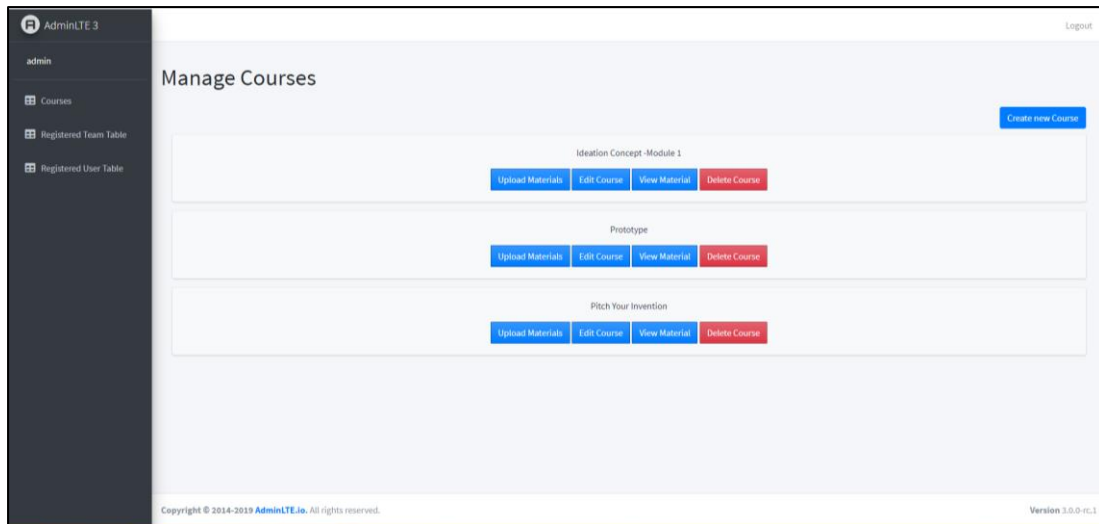


Figure 1.3 Admin Dashboard-Manage Course

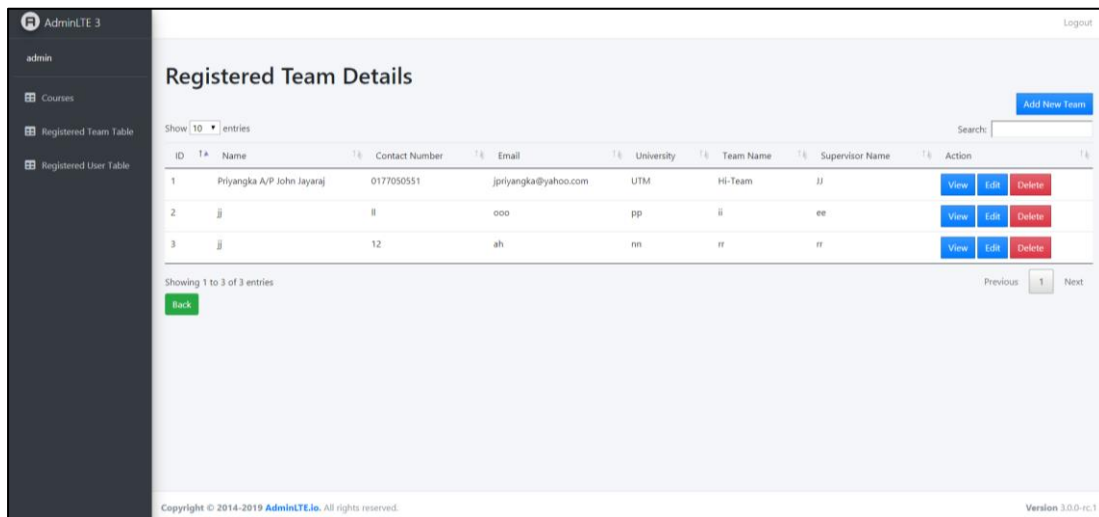


Figure 1.4 Admin Dashboard-Manage Registered Team

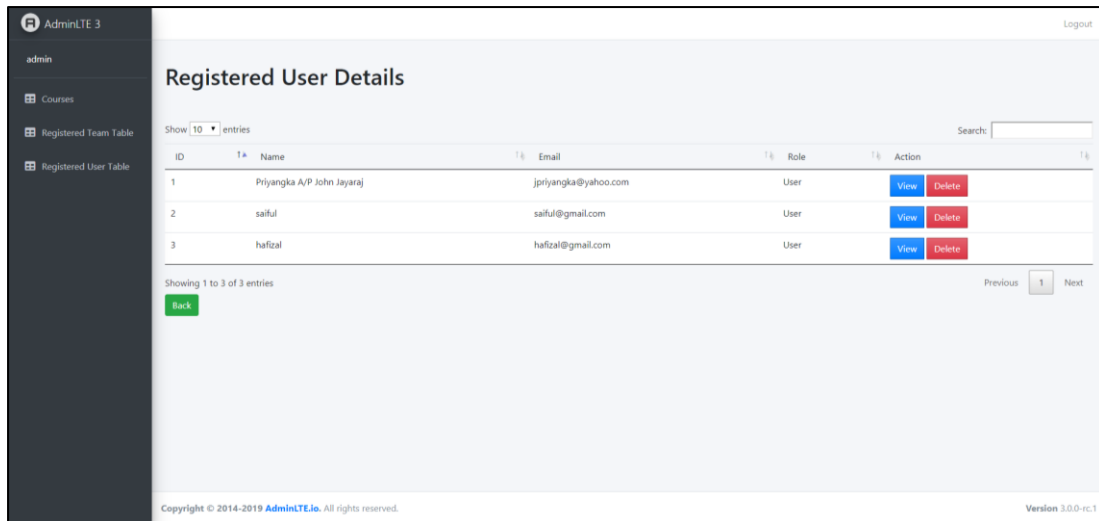


Figure 1.5 Admin Dashboard-Manage User Details

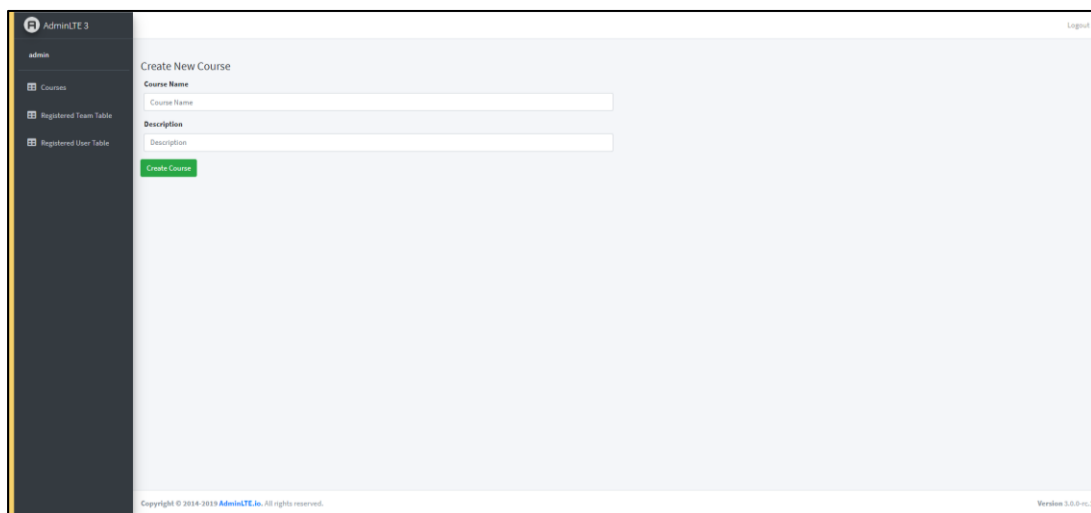


Figure 1.6 Admin Dashboard-Create New Course

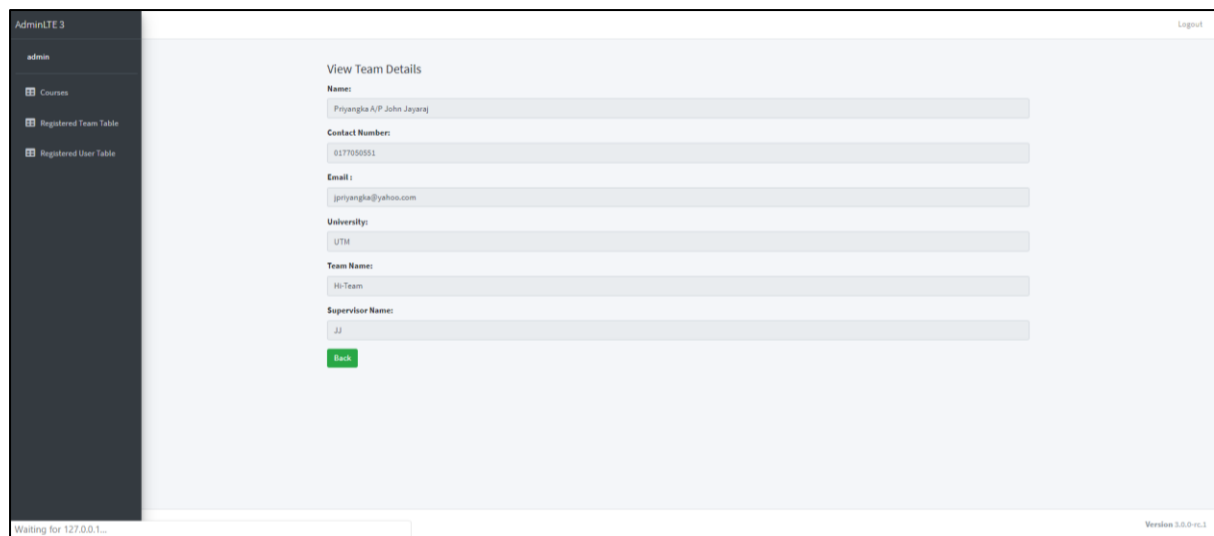


Figure 1.7 Admin Dashboard - View Team Details

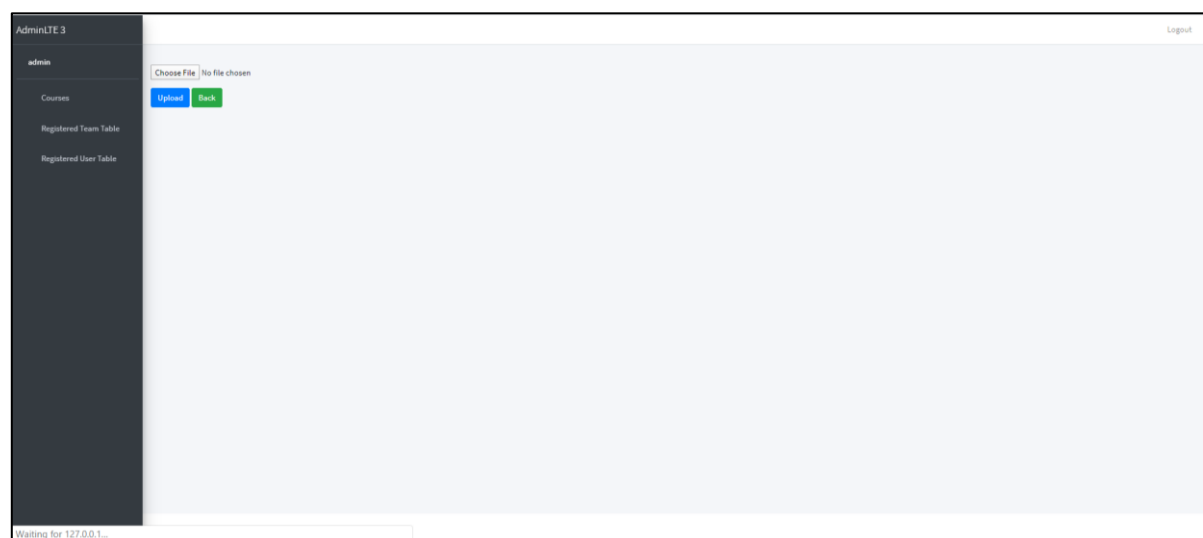


Figure 1.8 Admin Dashboard – Upload Materials

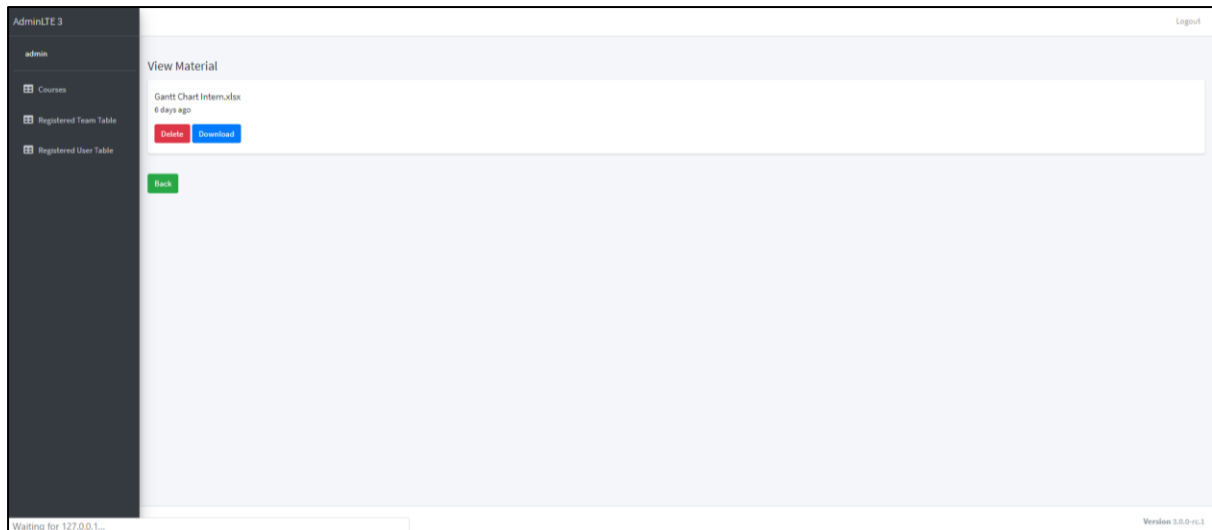


Figure 1.8 Admin Dashboard – View Materials

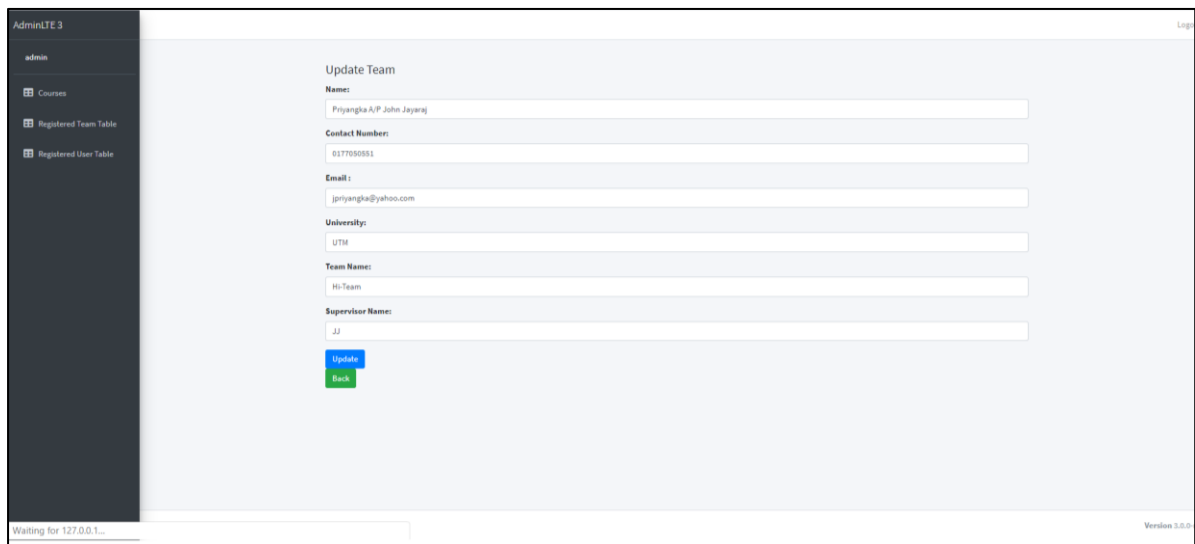


Figure 1.9 Admin Dashboard – Update Team Details

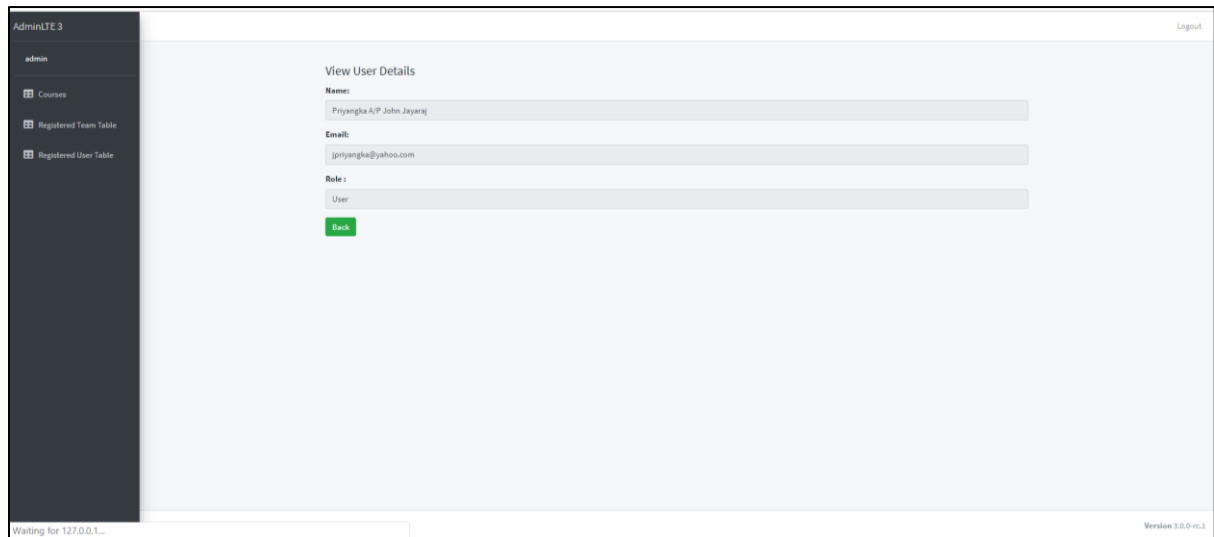


Figure 1.10 Admin Dashboard – View User Details