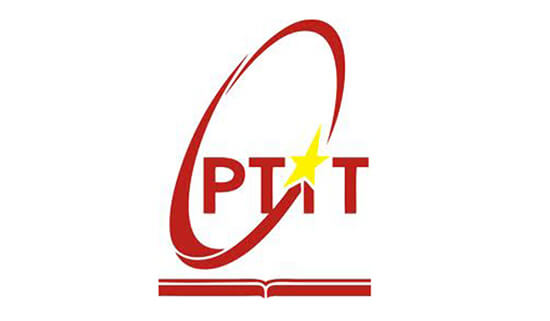
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA VIỄN THÔNG I**

----------



**BÁO CÁO**

**THỰC TẬP TỐT NGHIỆP**

**Giảng viên hướng dẫn : Nguyễn Thị Thu Hằng**

**Lê Tùng Hoa**

**Sinh viên thực hiện : Nguyễn Bá Trường**

**Mã sinh viên : B18DCVT432**

**Lớp : D18CQVT08-B**

**Hà Nội – 202****2**

# LỜI MỞ ĐẦU

Sau bốn năm là sinh viên thuộc khoa Viễn Thông 1, học trên giảng đường Học viện Công nghệ Bưu chính Viễn thông thì em đã tích lũy được cho bản thân mình nhiều kiến thức. Quá trình thực tập một mặt là yêu cầu bắt buộc với sinh viên nhưng mặt khác đây cũng là một giai đoạn hết sức ý nghĩa, quan trọng giúp sinh viên tập làm quen với công việc thực tế, hội nhập vào môi trường doanh nghiệp. Trong thời gian thực tập, sinh viên có thể học hỏi thêm được nhiều điều, đồng thời cũng là cơ hội để phát triển ra những điểm còn yếu trong kiến thức và khả năng của mình, để từ đó bổ sung, bù đắp trước khi rời ghế nhà trường.

Thực hiện nhiệm vụ học tập của nhà trường trong đợt thực tập tốt nghiệp, được sự đồng ý của ban lãnh đạo Khoa đào tạo và Công ty FPT Telecom. Bám sát đề cương của Khoa đưa ra, những lý luận và kiến thức đã học. Trong quá trình thực tập em đã được trau dồi thêm kiến thức về *lập trình, phát triển ứng dụng web* và hiểu biết hơn về ứng dụng cũng như cơ cấu và tình hình hoạt động của công ty.

Được sự tận tình giúp đỡ hướng dẫn của Công ty cùng với sự giúp đỡ của hai giảng viên hướng dẫn cô Nguyễn Thị Thu Hằng và cô Lê Tùng Hoa và nỗ lực trong học tập, tìm tòi học hỏi, em đã hoàn thành báo cáo thực tập tốt nghiệp của mình.

Báo cáo này gồm 2 phần chính:

* ***Phần I: Báo cáo thực tập tại công ty***
* ***Phần II: Báo cáo chuyên sâu***

Trong quá trình thực hiện báo cáo, tuy đã cố gắng nhưng em vẫn còn những hạn chế về thời gian tìm hiểu, kiến thức cũng như là kinh nghiệm và vẫn còn nhiều sai sót. Em rất mong được nhận những ý kiến đóng góp và nhận xét của các thầy cô để em có thể hoàn thiện hơn.

# LỜI CẢM ƠN

Để hoàn thành được báo cáo thực tập tốt nghiệp này thì em xin cảm ơn tới phía Công ty FPT Telecom đã tạo điều kiện cho em được thực tập tại đây. Ngoài ra, em cảm ơn anh hướng dẫn Nguyễn Thành Công và các anh trên công ty đã luôn nhiệt tình chỉ dẫn, giảng dạy cho em những kiến thức để em có thể nắm bắt nhanh nhất.

Em xin gửi lời cảm ơn tới ban lãnh đạo Học viện, Khoa đào tạo đã thiết lập khung chương trình có môn học “Thực tập” để em có thể vận dụng những kiến thức học được trên giảng đường áp dụng vào công việc và được tham quan, tìm hiểu về môi trường doanh nghiệp.

Em xin gửi lời cảm ơn sâu sắc và chân thành tới hai giảng viên hướng dẫn cô Nguyễn Thị Thu Hằng và cô Lê Tùng Hoa – hai cô luôn cống hiến hết mình với nghề, với sinh viên, luôn dìu dắt, giúp đỡ và chỉ bảo tận tình cho em từ lúc thực tập tới bây giờ.

Em chân thành cảm ơn!

Hà Nội, ngày 23 tháng 08 năm 2022

**MỤC LỤC**

[**LỜI MỞ ĐẦU 1**](#_Toc112152257)

[**LỜI CẢM ƠN 2**](#_Toc112152258)

[**PHẦN 1: THỰC TẬP TẠI CÔNG TY 5**](#_Toc112152259)

[**I. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP 5**](#_Toc112152260)

[**1.1. Giới thiệu về đơn vị thực tập 5**](#_Toc112152261)

[**1.2. Lịch sử hình thành và phát triển 5**](#_Toc112152262)

[**II. NỘI DUNG THỰC TẬP 5**](#_Toc112152263)

[**2.1. Django REST framework 5**](#_Toc112152264)

[**2.1.1. Tìm hiểu về Django REST Framework 5**](#_Toc112152265)

[**2.1.2. Cài đặt môi trường và khởi tạo Project 5**](#_Toc112152266)

[**2.1.3. Tạo một REST API 7**](#_Toc112152267)

[**2.1.4. Kiểm tra API với POSTMAN 11**](#_Toc112152268)

[**2.2. VueJs 15**](#_Toc112152269)

[**2.2.1. Tìm hiểu về Vuejs 15**](#_Toc112152270)

[**2.2.2. Cài đặt môi trường và khởi tạo Project 15**](#_Toc112152271)

[**2.2.3. Ứng dụng web với Vuejs và Django REST Framework 16**](#_Toc112152272)

[**PHẦN 2: BÁO CÁO CHUYÊN SÂU 20**](#_Toc112152273)

[**I. BÀI THỰC HÀNH SỐ 1 20**](#_Toc112152274)

[**1. Tìm hiểu về đo kiểm MS 20**](#_Toc112152275)

[**2. Máy đo 8922M 20**](#_Toc112152276)

[**3. Đo các đặc trưng của GSM 21**](#_Toc112152277)

[**3.1. Mục đích, yêu cầu 21**](#_Toc112152278)

[**3.2. Đo công suất đỉnh sóng mang 21**](#_Toc112152279)

[**3.3. Đo lỗi pha và tần số 21**](#_Toc112152280)

[**3.4. Mặt nạ phổ công suất 22**](#_Toc112152281)

[**II. BÀI THỰC HÀNH SỐ 2 23**](#_Toc112152282)

[**1. Sử dụng phần mềm Tems Pocket 23**](#_Toc112152283)

[**1.1. Mục đích, yêu cầu 23**](#_Toc112152284)

[**1.2. Bài đo Ping 23**](#_Toc112152285)

[**1.3. Bài đo Idle 24**](#_Toc112152286)

[**1.4. Bài đo CSFB 24**](#_Toc112152287)

[**1.5. Bài đo ứng dụng YouTube và các thông số RSRP, RSRQ, RSSI, … 25**](#_Toc112152288)

[**2. Sử dụng phần mềm Tems Discovery 26**](#_Toc112152289)

[**2.1. Mục đích, yêu cầu 26**](#_Toc112152290)

[**2.2. Đọc và phân tích logfile bằng Tems Discovery 26**](#_Toc112152291)

[**KẾT LUẬN CHUNG 29**](#_Toc112152292)

# PHẦN 1: THỰC TẬP TẠI CÔNG TY

1. **TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP**
   1. **Giới thiệu về đơn vị thực tập**

Công ty Cổ phần Viễn thông FPT (FPT Telecom) là một trong những nhà cung cấp dịch vụ Viễn thông và Internet hàng đầu khu vực.

Trụ sở chính: Tòa nhà FPT Tower, số 10 Phạm Văn Bạch, Cầu Giấy, Hà Nội.

* 1. **Lịch sử hình thành và phát triển**

FPT Telecom được thành lập ngày 31/01/1997, được khởi nguồn từ Trung tâm Dịch vụ Trực tuyến do 4 thành viên sáng lập cùng sản phẩm internet đầu tiên của Việt Nam mang tên “Trí tuệ Việt Nam – TTVN”, sản phẩm được coi là đặt nền móng cho sự phát triển của internet tại Việt Nam.

FPT Telecom là một thành viên của tập đoàn FPT - một trong những công ty dịch vụ *công nghệ thông tin* lớn nhất tại Việt Nam với lĩnh vực kinh doanh chính là cung cấp các sản phẩm dịch vụ công nghệ thông tin.

Sau hơn 24 năm hoạt động, FPT Telecom đã có hơn 9500 nhân viên chính thức với gần 300 văn phòng giao dịch thuộc gần 90 chi nhánh và có mặt trên 59 tỉnh thành trên cả nước. Bên cạnh đó công ty đã và đang đặt dấu ấn trên trường quốc tế.

1. **NỘI DUNG THỰC TẬP**
   1. **Django REST framework**
      1. **Tìm hiểu về Django REST Framework**

Trong bối cảnh phát triển ứng dụng web, chúng ta thường nói về RESTful API. Đây là một cách phổ biển để cung cấp ứng dụng cho bên thứ ba (các ứng dụng và web). Điều này chỉ cần cung cấp thông tin lưu trữ trong cơ sở dữ liệu ở định dạng phổ biến như XML hoặc JSON, bằng cách này một ứng dụng bên thứ ba có thể tương tác với dữ liệu mà không cần kết nối trực tiếp với cơ sở dữ liệu, cũng vì thế không quan trọng cơ sở dữ liệu là MySQL hay PostgreSQL, hoặc nếu ứng dụng được viết bằng Java hoặc Python, nhưng RESTful APIs có thể được sử dụng để sửa chữa cơ sở dữ liệu.

Django REST framework là một công cụ hỗ trợ đắc lực trong việc xây dựng WebAPI. Một số lý do khi ta nên sử dụng REST:

* Có hỗ trợ dữ liệu ORM và non-ORM với Serialization.
* Tài liệu phong phú, cộng đồng hỗ trợ lớn.
* Được sử dụng trong các công ty quốc tế như Mozilla, Red Hat, Heroku và Eventbrite.
  + 1. **Cài đặt môi trường và khởi tạo Project**
* **Cài đặt môi trường:**

Tạo thư mục dự án:

D:\>mkdir demoDjango

D:\>cd demoDjango

Tạo ra một môi trường ảo (virtual environment) để làm việc:

D:\demoDjango>py -m venv demoEnv

D:\demoDjango>demoEnv\Scripts\activate

(demoEnv) D:\demoDjango>

Tiếp theo là install Django và django-rest-framework:

(demoEnv) D:\demoDjango>pip install django

(demoEnv) D:\demoDjango>pip install djangorestframework

* **Tạo một project và app:**

Tạo một project (myproject) và một app (task):

(demoEnv) D:\demoDjango>django-admin startproject myproject

(demoEnv) D:\demoDjango>cd myproject

(demoEnv) D:\demoDjango\myproject>py manage.py startapp task

#Khởi tạo dữ liệu ban đầu cho project

(demoEnv) D:\demoDjango\myproject>py manage.py migrate

Chạy thử lần đầu:

(demoEnv) D:\demoDjango\myproject>py manage.py runserver

Truy nhập đường dẫn: <http://127.0.0.1:8000/>

Như hình bên dưới, chúng ta đã khởi tao một ứng dụng web thành công.

Graphical user interface, text, application, website

Description automatically generated

Tiếp theo ta thêm rest\_framework và task vào setting.py

INSTALLED\_APPS = (

...

‘rest\_framework’,

‘task’

)

* + 1. **Tạo một REST API**

Đầu tiên chúng ta sẽ tìm hiểu qua về các thành phần cụ thể:

* admin.py: File config trong trang admin của app.
* model.py: Làm việc với database. Nơi tạo dữ liệu, các trường, các bảng.
* serializer: Chuyển queryset hoặc model của django thành dạng dữ liệu dễ dàng render trên web hơn và ngược lại, chuyển JSON/XML do client chuyển lên thành object để Django có để xử lý dễ dàng.
* views.py: views sẽ thể hiện dữ liệu trả về cho người dùng, nó không chỉ là cách hiển thị dữ liệu như thế nào, mà còn là những dữ liệu nào được hiển thị nữa.
* urls.py: Các đường dẫn.

Trong file models.py tạo 2 model có tên là User và Task :

from django.db import models

class User(AbstractUser):

first\_name = models.CharField(max\_length=100)

last\_name = models.CharField(max\_length=100)

username = models.CharField(max\_length=100 , unique=True, primary\_key=True)

email = models.EmailField(max\_length=200)

password = models.CharField(max\_length=100)

REQUIRED\_FIELDS = []

def \_\_str\_\_(self):

return self.username

class Task(models.Model):

entrustor = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='entrustor')

assign = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='assign')

title = models.CharField(max\_length=100, blank=True, default='')

content = models.TextField()

estimate = models.DateTimeField(blank=True)

note = models.CharField(max\_length=255, blank=True, default='')

created\_date = models.DateTimeField(auto\_now\_add=True, editable=False)

finish\_date = models.DateTimeField(blank=True, null=True)

modified\_date = models.DateTimeField(auto\_now=True, blank=True)

def \_\_str\_\_(self):

return self.title

Thay đổi database và apply thay đổi bằng lệnh :

#(demoEnv) D:\demoDjango\myproject>

py manage.py makemigrations

py manage.py migrate

# Tạo admin user:

py manage.py createsuperuser

# Chạy trương trình:

py manage.py runserver

Trong file serializers.py tạo 2 class UserSerializer và TaskSerializer.

from rest\_framework import serializers

from .models import Task, User

class UserSerializer(serializers.ModelSerializer):

class Meta:

model = User

fields = ('username', 'email', 'password')

class TaskSerializer(serializers.ModelSerializer):

class Meta:

model = Task

fields = ('id', 'entrustor', 'assign', 'title', 'content', 'estimate', 'note', 'created\_date', 'finish\_date', 'modified\_date')

File views.py:

from django.views.decorators.csrf import csrf\_exempt

from django.http.response import JsonResponse

from rest\_framework.parsers import JSONParser

from rest\_framework.views import APIView

from .models import Task, User

from .serializers import TaskSerializer, UserSerializer

@csrf\_exempt

def taskApi(request,pk=0):

if request.method == 'GET':

try:

if pk != None:

task = Task.objects.get(id = pk)

tasks\_serializer = TaskSerializer(task, many = False)

except:

tasks = Task.objects.all() # Lay tat ca task

tasks\_serializer = TaskSerializer(tasks, many = True)

return JsonResponse(tasks\_serializer.data, safe = False)

elif request.method == 'POST':

task\_data = JSONParser().parse(request) # phan tich thong tin den

tasks\_serializer = TaskSerializer(data=task\_data)

if tasks\_serializer.is\_valid(): # Neu hop le -> luu

tasks\_serializer.save()

return JsonResponse("Added Successfully", safe = False)

return JsonResponse("Failed to Add", safe=False)

elif request.method == 'PUT':

task\_data = JSONParser().parse(request)

task = Task.objects.get(id = task\_data['id'])

tasks\_serializer = TaskSerializer(task, data=task\_data)

if tasks\_serializer.is\_valid():

tasks\_serializer.save()

return JsonResponse("Update Successfully", safe = False)

return JsonResponse("Failed to Update", safe=False)

elif request.method == 'DELETE':

task = Task.objects.get(id = pk)

task.delete()

return JsonResponse("Deleted Successfully", safe=False)

class UserView(APIView):

def get(self, request):

user = User.objects.all() # Lay tat ca task

user\_serializer = UserSerializer(user, many = True)

return JsonResponse(user\_serializer.data, safe = False)

def post(self, request):

user\_data = JSONParser().parse(request)

user\_serializer = UserSerializer(data=user\_data)

if user\_serializer.is\_valid():

user\_serializer.save()

return JsonResponse("Added Successfully", safe = False)

return JsonResponse("Failed to Add", safe=False)

Trong 2 file serializers.py và views.py này sẽ code các thao tác để tạo được API.

Để API có thể hoạt động ta phải gán nó cho một url nào đó. File urls.py:

from django.urls import path

from .views import UserView

from task import views

urlpatterns = [

path('user/', UserView.as\_view()),

path('task/', views.taskApi),

path('task/<int:pk>/', views.taskApi),

]

Kiểm tra trên web với đường dẫn: <http://127.0.0.1:8000/task/> ta nhận được kết quả hiển thị là các chuỗi JSON có dạng như hình dưới.

Graphical user interface, text

Description automatically generated

* + 1. **Kiểm tra API với POSTMAN**

**Giới thiệu về POSTMAN:**

* POSTMAN là một công cụ cho phép chúng ta làm việc với API, nhất là REST. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào. Làm chủ Postman, bạn sẽ thấy việc gọi các Rest API (như Facebook, Google, Youtube) chả có gì phức tạp cả.
* POSTMAN hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, ...).
* POSTMAN cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

Method GET để gọi API:

**Graphical user interface, text, application, email

Description automatically generated**

Method POST thêm nội dung vào phần body:

**Graphical user interface, text, application, email

Description automatically generated**

Kiểm tra lại với Method GET:

Graphical user interface, text, application, email

Description automatically generated

Method PUT: Update dữ liệu

Graphical user interface, text, application, email

Description automatically generated

Kiểm tra lại:

Graphical user interface, text, application, email

Description automatically generated

Dữ liệu đã được cập nhập, ngày update cũng đã được cập nhập.

Method DELETE:

Graphical user interface, text, application

Description automatically generated

Kiểm tra lại:

Graphical user interface, text, application, email

Description automatically generated

Xóa thành công.

* 1. **VueJs** 
     1. **Tìm hiểu về Vuejs**

Vue.js là một framework Javascript được tạo bởi Evan You, giúp chúng ta xây dựng giao diện người dùng cũng như xây dựng Single Page Application thân thiện với người dùng, chúng xây dựng từ các thư viện, cách triển khai component, các chức năng đặc trưng của nó như SFC (Single File Component). Phiên bản ổn định mới nhất hiện tại của Vue.js là 2.6.10. Nào chúng ta cùng đi vào những kiến thức cơ bản nhất của Vue.

* + 1. **Cài đặt môi trường và khởi tạo Project**

Cài đặt môi trường:

npm install -g @vue/cli

Tạo Project:

vue create myproject\_vue3

cd myproject\_vue3

npm run serve

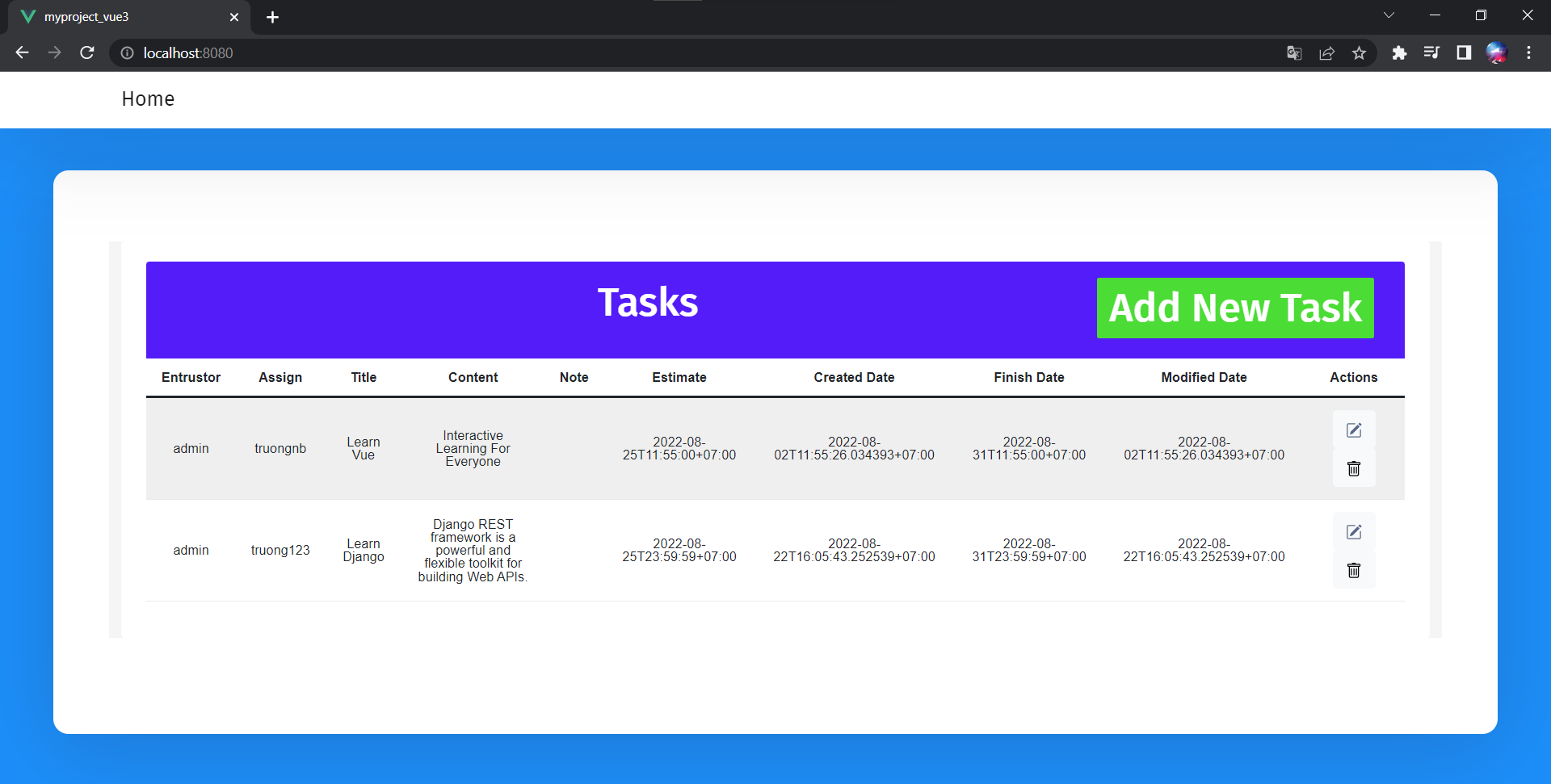
Khỏi tạo thành công:

Logo

Description automatically generated with low confidence

* + 1. **Ứng dụng web với Vuejs và Django REST Framework**

Màn hình ứng dụng web sau khi request Get dữ liệu từ API.



Thêm nhiệm vụ vào Tasks bằng cách click vô Add New Task, màn hình hiển thị:

Graphical user interface

Description automatically generated

Kết quả: Nhiệm vụ mới đã được thêm vô Tasks

A screenshot of a computer

Description automatically generated with medium confidence

Update nhiệm vụ, click vô chỉnh sửa:

Graphical user interface, website

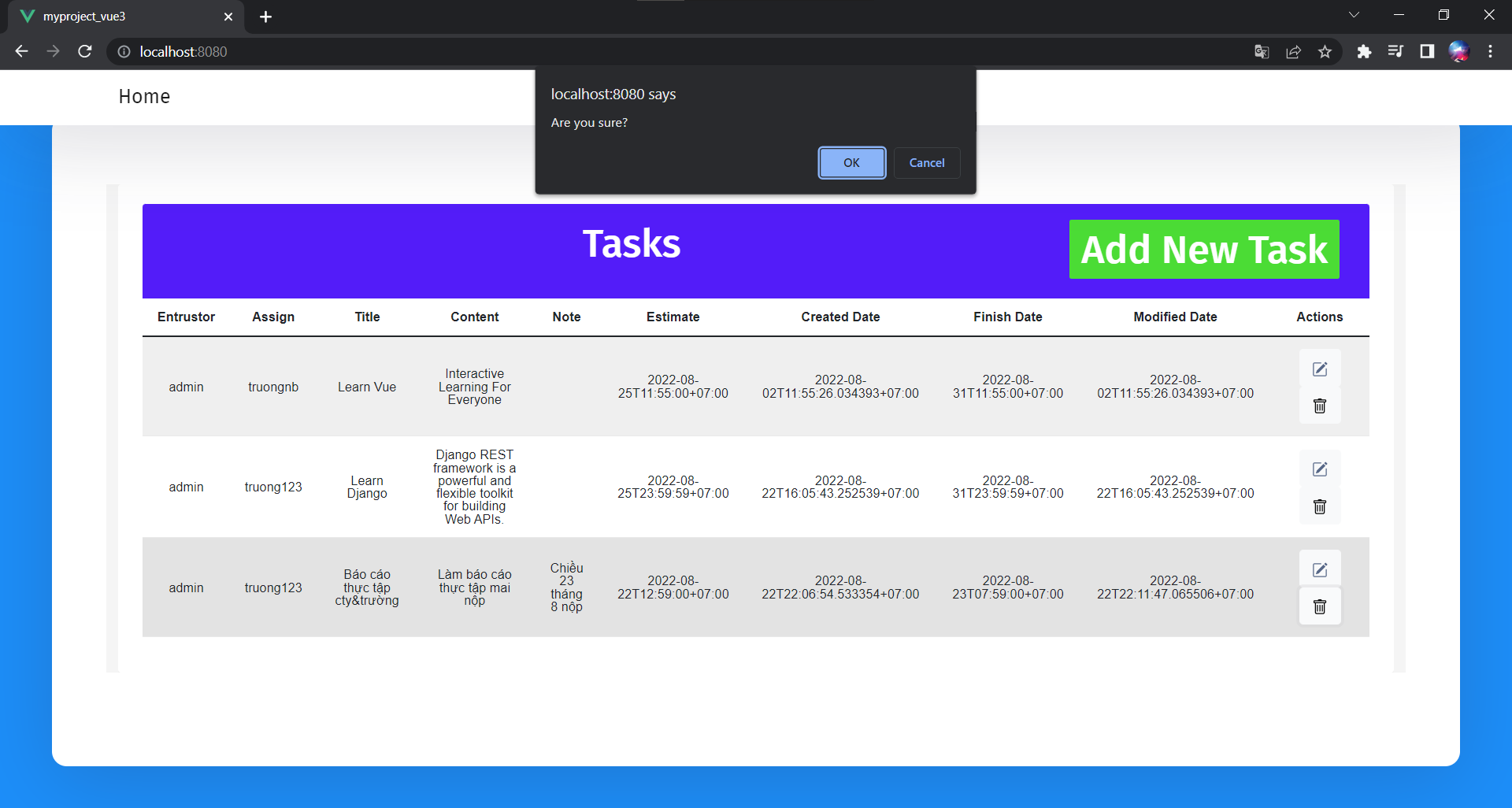
Description automatically generated

Kết quả: Update thành công

Graphical user interface, application, Word

Description automatically generated

Xóa nhiệm vụ, click vô icon thùng giác:



Kết quả:

Graphical user interface, application

Description automatically generated

# PHẦN 2: BÁO CÁO CHUYÊN SÂU

1. **BÀI THỰC HÀNH SỐ 1**
2. **Tìm hiểu về đo kiểm MS**

Cấu trúc MS:

**Diagram

Description automatically generated**

Hình 1: Sơ đồ khối của một MS

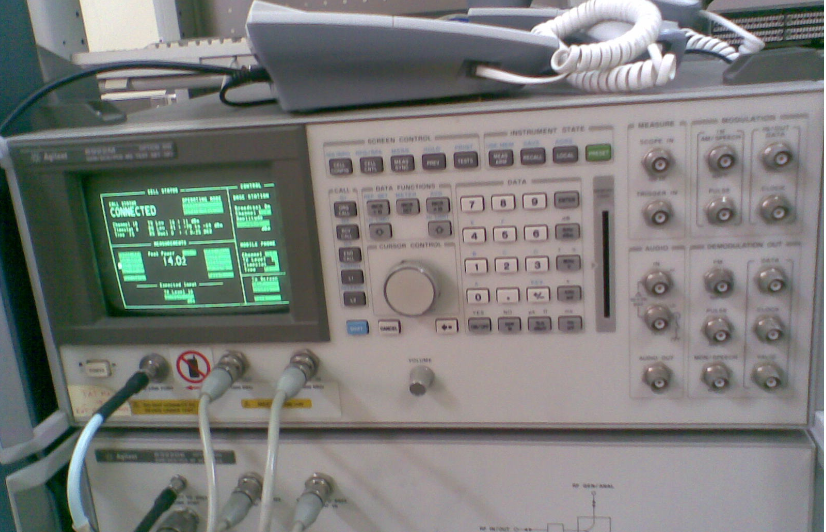
MS gồm hai phần:

* Phần vô tuyến. Thực hiện các chức năng phát, thu, điều chế, giải điều chế
* Phần xử lý số. Thực hiện các chức năng giao tiếp tín hiệu, xử lý số điều khiển và báo hiệu.

Hai phần này được tích hợp chung trên một tấm mạch in, cùng với một tấm mạch in cho giao tiếp với người dùng bao gồm bàn phím và màn hiển thị.

1. **Máy đo 8922M**

* Agilent 8922M GSM Test Setlà máy đo, kiểm tra các thông số của trạm di động MS theo tiêu chuẩn GSM.
* Đặc trưng của GSM đối với máy di động MS là yêu cầu cần thiết để đánh giá chất lượng của MS.
* Máy có thể làm việc ở 3 chế độ: Ô tích cực (Active cell), chế độ kiểm tra (Test mode), tạo sóng mang liên tục (CW-Generator).
* Máy làm việc ở băng tần 900MHz.
* 8922M kết hợp với Opt 010: 83220E tạo nên 8922P- Hệ thống kiểm tra đa băng tần.
* Máy có thể thực hiện các phép đo cơ bản: Công suất sóng mang đỉnh, lỗi pha và lỗi tần số, mặt nạ công suất, BER, phổ tín hiệu RF đầu ra.
* Máy có thể được định cấu hình làm việc như: Máy phân tích phổ, máy hiện sóng, máy đo âm tần, máy đo sóng mang liên tục.



Hình 3.2. Máy đo kiểm Aligent 8922M

1. **Đo các đặc trưng của GSM**
   1. **Mục đích, yêu cầu**

* Bài đo thực hiện các phép đo cơ bản xác định đặc trưng của hệ thống GSM: Công suất sóng mang đỉnh; Lỗi pha và tần số; Mặt nạ phổ công suất.
* Các phép đo được thực hiện trên máy đo GSM “Aligent Technologies 8922M” bằng việc lựa chọn các trường trong vùng “Measurements” ở màn hình Cell Controll.
* Để thực hiện phép đo, cần thiết lặp cuộc gọi giữa MS và 8922M.
  1. **Đo công suất đỉnh sóng mang**

Mục đích: Thực hiện đo và tính trung bình công suất sóng mang máy phát cho một cụm đơn.

Kết quả: Tờ báo cáo “GSM-MS TEST REPORT” mục I.

* 1. **Đo lỗi pha và tần số**

Mục đích: Xác định giá trị lỗi pha và tần số của tín hiệu do quá trình điều chế và tạp âm đối với các hướng phát phần vô tuyến.

Kết quả: Tờ báo cáo “GSM-MS TEST REPORT” mục II.

* 1. **Mặt nạ phổ công suất**

Mục đích: Phép đo hiển thị xung tín hiệu phát của MS trong khoảng thời gian một cụm và kiểm tra xem mức này có phù hợp với các chuẩn GSM hay không.

Để tránh nhiễu không mong muốn và đảm bảo thu tốt ở trạm gốc BTS, tín hiệu được phát ở MS phải tuân theo các chuẩn GSM.

Kết quả: Tờ báo cáo “GSM-MS TEST REPORT” mục III

1. **BÀI THỰC HÀNH SỐ 2**
   1. **Sử dụng phần mềm Tems Pocket**

TEMS Pocket là một chương trình hỗ trợ đo sóng di động chạy trên hệ điều hành Android.

Các tính năng được hỗ trợ công tác đo kiểm:

* Hỗ trợ đo các tín hiệu của các hệ GSM, WCDMA, LTE, Wi-fi.
* Các chế độ Data view gồm: Idle, Dedicated, Data, Test Status, Location.
* Hỗ trợ quét sóng đo vùng phủ trên map online (outdoor mode) với GPS tích hợp sẵn trên thiết bị hoặc GPS rời.
* Chức năng scanning cần phải có thiết bị quét sóng chuyên dụng. Cụ thể là DRT hoặc PCTel.
* Hỗ trợ đo vùng phủ indoor mode (pinpoint).
* Các chức năng đo kiểm và trãi nghiệm của người dùng như voice, SMS,
* FTP, Ping, Streaming (Youtube), Tweeter, Facebook, HTTP. Các chức năng này được thực hiện thông qua các script.
* Hỗ trợ lock RAT, lock band, lock cell.
* Hỗ trợ ghi lại logfile của quá trình đo kiểm phục vụ cho việc phân tích xử lý hậu kỳ.
  1. A screenshot of a computer

     Description automatically generated with medium confidence**Mục đích, yêu cầu**
* Để sinh viên nám được các sử dụng phần mềm TEMS POCKET để đo các KPI của 4G trong thực tế, cho sinh viên có cái nhìn khái quát trong việc làm hiện nay về công việc đo kiểm.
* Đánh giá được chất lượng của dịch vụ 4G.
  1. **Bài đo Ping**

**Bước 1:** Vào menu script => chọn New Action => Logfile recording.

**Bước 2:** Phần Logfile tag (đặt tên Logfile để phân biệt giữa các dịch vụ với nhau).

**Bước 3:** Chọn New Action => Ping, điền các thông số.

**Bước 4:** Chọn New Action => Wait (thời gian chờ).

Sau khi cài đặt xong, ấn Start Script.

* 1. A screenshot of a video game

     Description automatically generated**Bài đo Idle**

Các bước thiệt lập làm tương tư như bài đo Ping, nhưng thay vì ở *bước* *3* mình chọn Ping thì ở bài này mình chọn Youtube. Sau đó ta thực hiện bài đó bằng cách:

**Bước 1:** Bật chế độ máy bay.

**Bước 2:** Chạy bài đó Idle vừa tạo.

**Bước 3:** Tắt chế độ máy bay.

**Bước 4:** Chờ video load.

* 1. **Bài đo CSFB**

Các bước thiệt lập làm tương tư như bài đo Ping, nhưng thay vì ở *bước* *3* mình chọn Ping thì ở bài này mình chọn Voice MO.

Trong Voice MO ta nhập số điện thoại gọi đến vào đây, ở bài đó này chúng ta sẽ gọi điện đến 2 số: Số máy cùng mạng và ngoại mạng.

Sau khi Start script ta thu được kết quả:

**Gọi nội mạng:**

Graphical user interface

Description automatically generated with low confidence Graphical user interface

Description automatically generated with low confidence Graphical user interface

Description automatically generated

Màn hình đang gọi Màn hình đang bắt máy Màn hình khi kết thúc cuộc gọi

**Gọi ngoại mạng:**

Graphical user interface

Description automatically generatedGraphical user interface, text

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Màn hình đang gọi Màn hình đang bắt máy Màn hình khi kết thúc cuộc gọi

Các thông số:

* Số bị gọi: 0372720373(nội mạng), 0941216898(ngoại mạng)
* A screenshot of a computer

  Description automatically generated with medium confidenceDomain: CS (Circuit Switching)
* Directon: MO (Mobile Original) - thực hiện gọi đi
* Call State:
* Setup – cuộc gọi đang được thiết lập
* Established - cuộc gọi đã được thiết lập.
* Setup Time: thời gian thiết lập là 4.120 giây
* Call Duration: là thời gian cuộc gọi đang diễn ra
* Cfg Duration: là thời gian được cấu hình trên script cho cuộc gọi. Ở đây là 30 giây.
* Speech Codec:
  + Nội mạng: EFR
  + Ngoại mạng: AMR
  1. **Bài đo ứng dụng YouTube và các thông số RSRP, RSRQ, RSSI, …**

Thiết lập thông số: Vào menu => Custom top View => LTE

Với mỗi ô trong bảng ta click vô chọn label để đặt tên, chọn value bar hoặc value label chọn các giá trị.

1. **Sử dụng phần mềm Tems Discovery**

* Tems Discovery là phần mềm Post-Processing
* TD là công cụ phân tích tối ưu mạng
* Được nghiên cứu và sản xuất bởi hãng Ascom
* Tems Discovery Device là phần mềm chuyên dụng cho việc phân tích và làm báo cáo kết quả đó kiểm
* Việc quản lý dữ liệu TD thông qua các Project, có 2 loại project: private và public. Mỗi project bao gồm các Datasets và Composite Dataset. Mỗi Dataset chứa dữ liệu logfile dễ dàng tổ chức File name, Operator, IMSI, …
  1. **Mục đích, yêu cầu**
* Tạo project trên TDD
* Cách import, export dữ liệu, query, celfile, plotband, route GCF, … và import logfile nhận được từ team đo kiểm cho việc làm report, phân tích trên TDD.
* Tạo composite để gộp những data riêng lẻ của một dịch vụ hay một MS nào đó thành một thể thống nhất.
* Yêu cầu: Sinh viên nắm được cách sử dụng phầm mềm Tems Discovery
  1. **Đọc và phân tích logfile bằng Tems Discovery**

Thông tin bản tin báo hiệu bắt được tín hiệu LTE NAS từ logfie trên điện thoại ở phần 1

Graphical user interface, text, application, email

Description automatically generated

Thông tin bản tin báo hiệu bắt được tín hiệu LTE RRC từ logfile trên điện thoại ở phần 1

Graphical user interface, text, application, email

Description automatically generated

Phân tích chi tiết bản tin báo hiệu DL-DCCH của tín hiệu LTE RRC

Graphical user interface, application

Description automatically generated

Phân tích chi tiết bản tin báo hiệu UL-CCCH của tín hiệu LTE RRC

Graphical user interface, text, application, email

Description automatically generated

# KẾT LUẬN CHUNG

Sau thời gian 6 tuần: gồm 4 tuần thực tập tại công ty FPT Telecom và 2 tuần nghiên cứu chuyên sâu ở trường, với sự nỗ lực, cố gắng của bản thân em cùng với sự hướng dẫn tận tình của cô Nguyễn Thị Thu Hằng và cô Lê Tùng Hoa cũng như sự giúp đỡ từ các anh chị trong công ty thì em đã hoàn thành báo cáo thực tập tốt nghiệp.

Tuy nhiên cũng phải nói rằng dù đã cố gắng rất nhiều, nhưng em vẫn không tránh khỏi những sai sót trong việc xây dựng và thiết kế hệ thống cũng như là cách thức trình bày một báo cáo thực tập, vì vậy em rất mong nhận được sự chỉ bảo và đóng góp ý kiến của thầy/cô để báo cáo của em hoàn thiện hơn.