Final Report

Emergency Web-Application

Team 6

Nont Arayarungsarit st124335

Nuon Puthisoptey st124390

Present to

Professor Chantri Polprasert

This report is part of Full Stack Application Development course

Asian Institute of Technology (AIT)

Content

1.Introduction

In an era marked by unprecedented challenges, ensuring the safety and well-being of students and staff within educational institutions is of paramount importance. Recognizing the need for a proactive and efficient approach to emergency response. Nowadays, There are many accident notifications and information on social media with community-driven but often too broadly and widely. While it has the advantage of providing users with a wide range of information, the downside is that sometimes it's challenging to focus on what one specifically needs to know so we created the website which can focus only specific news in our university. Our group propose the development and implementation of an emergency university website application. This innovation development aims to empower users, ranging from students, faculty, and staff, to swiftly and effectively report emergencies, fostering a secure learning environment.

Our application derives the types of emergency into 5, including harassment, fire accidents, accidental, medical, and others. Our application provides timely and accurate reporting based on real-time submission and notifications of the report in order to foster for a rapid and effective response. Furthermore, facilitating effective communication is our priority. Therefore, the application is equipped with comments that allow users to exchange updates. Notifications and updates will be disseminated to relevant parties, fostering a collaborative and coordinated effort.

The emergency university website application aims to create a sense of community responsibility by encouraging all stakeholders to actively participate in the safety of the AIT university environment.

2.Problem statement

Despite the increasing focus on school safety, many educational institutions face challenges in establishing a swift and effective response system for emergencies. Traditional methods of reporting incidents often suffer from delays, inaccuracies, and lack of standardized procedures, hindering the ability of schools to promptly address critical situations. As a result, our group establish this user-friendly emergency reporting solution that can streamline the reporting process, enhance communication, and empower stakeholders to contribute actively to the safety of the school environment.

Several pressing problems in regards to the traditional methods of reporting emergencies include:

2.1Inefficiencies in reporting : filing paper-based form in order to submit an emergency appears to be outdated, and are prone to inefficiencies and time consumption.

2.2 Communication gaps : the communication gap between individuals reporting emergencies and the designated emergency response teams can lead to misunderstanding, resulting in delayed or ineffective responses. Clear communication is essential for a coordinated effort during emergencies.

2.3 Limited community involvement : our school, AIT, has enough resource to implement this project, however, we lack the tools to engage the entire school in the safety process. Hence, a comprehensive emergency reporting system, such as this one, would encourage participation from students, faculty, staff, and parents to create a collective and vigilant approach to school security.

In light of these challenges, the proposed Emergency School Website Application aims to address these critical issues and revolutionize the way schools handle emergency reporting, ensuring a safer and more responsive educational environment for all stakeholders.

3.Related works

Several related works and applications exist that address the need for improved emergency reporting and response systems in educational institutions. While the specific features and functionalities may vary, these initiatives share a common goal of enhancing school safety.

3.1 Safe Schools Alert: an online platform designed to empower students, staff, and parents to report safety concerns, including bullying, harassment, and other potential threats. It offers features such as anonymous reporting and incident tracking.

3.2 Button: an emergency alert system that allows individuals to quickly notify authorities about various emergencies, including school incidents. It many focus on geolocation services.

3.3 School Dude Crisis Manager: a mobile application designed to assist schools in emergency preparedness and response. It provides tools for creating and accessing emergency plans, communicating with stakeholders, and reporting incidents.

While these existing applications share common objectives, our proposed Emergency School Website Application can distinguish itself by offering a tailored and comprehensive solution that addresses the specific and most common needs at AIT. These related works has helped us to identify the best practices, and come up with features that responds to the challenges to incorporate into the proposed application.

4.Methodology

For the methodology, We have define and plan some steps begin with Development, Testing and Deployment.

4.1 System Architecture

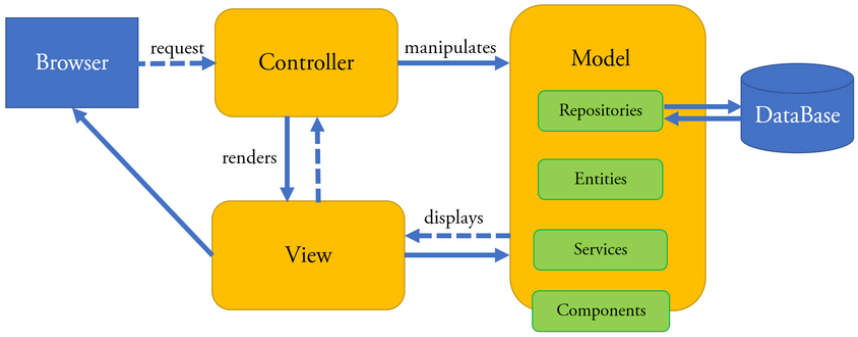


Figure 4.1 : Our website that based on Model-View-Controller (MVC) Architecture

Image source : https://www.researchgate.net/figure/The-Spring-MVC-architecture-as-depicted-in-16_fig5_349049076

An architectural pattern commonly used in software development for creating user interfaces. It separates an application into three interconnected components

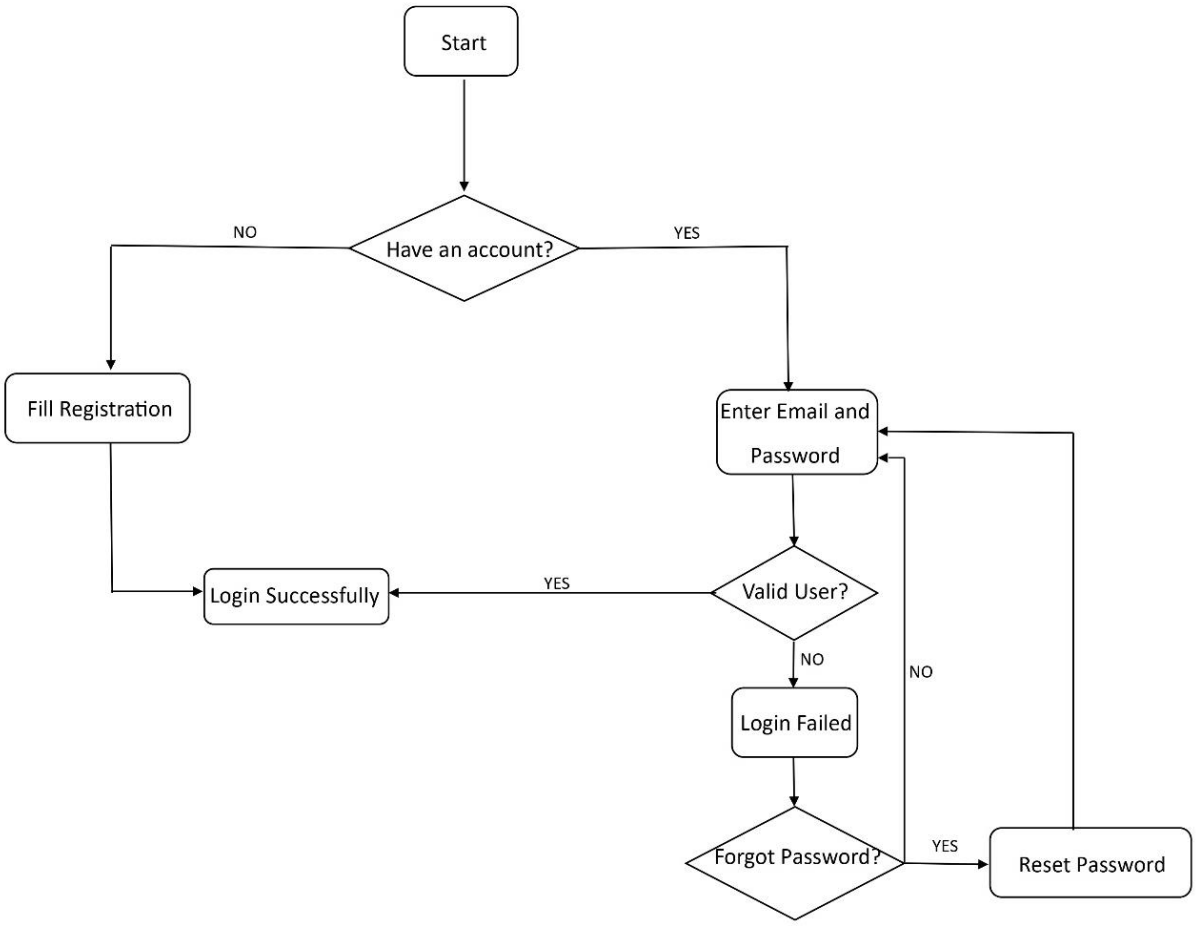
4.1.1 Model is a component represents the data and the business logic of the application. It manages the data, responds to queries about that data, and performs the necessary actions. Essentially, the model manages the application's data and business rules.

4.1.2 View The view is responsible for presenting the data to the user. It's the user interface that displays the information from the model to the user and interacts with the user's actions, like clicking buttons or entering data into forms.

4.1.3 Controller Controllers act as an intermediary between the model and the view. They handle user input, manipulate the model based on that input, and update the view accordingly. They process incoming requests, work with the model to perform any required operations, and render the appropriate view to the user.

The main idea behind MVC is to separate the concerns of an application, making it easier to manage, maintain, and extend. It promotes a clean separation of logic and presentation, allowing developers to work on different components independently.

Our group, Team 6, has log-in and register flowchart as follows



4.2 Testing Result can help to test some functionality of the web-application. In this case we will use cucumber for BDD (Behavior Driven Development) and we will assume to create user that has account for admin because admin can have more role to than normal user.

```
$ rails db:seed
$ bundle exec cucumber
Using the default profile...
```

```
Feature: Links
  Scenario: Get course info                                    # features/links.featu
re:3
    User who has account should be able to see the list of comment working on post
    Given I am user and admin at the same time                 # features/step_defini
tions/link_steps.rb:1
    Given There is a post                                      # features/step_defini
tions/link_steps.rb:7
    And post have some comments                                # features/step_defini
tions/link_steps.rb:12
    And I am logged in to access website                       # features/step_defini
tions/link_steps.rb:17
    When I visit the landing page                              # features/step_defini
tions/link_steps.rb:22
    Then I should see the list of emergency                    # features/step_defini
tions/link_steps.rb:27
    When I click comment to see the comment                    # features/step_defini
tions/link_steps.rb:35
Checking for expected text of nil is confusing and/or pointless since it will alway
s match. Please specify a string or regexp instead. /builds/ait_fsad-2023/team-6/em
ergency-webapp/features/step_definitions/link_steps.rb:42
```

```
    Then The page should show some of comment on that post     # features/step_defini
tions/link_steps.rb:40
    When I visit the land                                      # features/step_defini
tions/link_steps.rb:50
    Then I should see the button which only for admin          # features/step_defini
tions/link_steps.rb:56
    And I click only admin button                              # features/step_defini
tions/link_steps.rb:61
    Then tha page directly through Admin Zone page             # features/step_defini
tions/link_steps.rb:66
    And I should see the list of post and user management      # features/step_defini
tions/link_steps.rb:72
    Then I click list of user to see user who has registered # features/step_defini
tions/link_steps.rb:78
    And I can see list of them and number of users             # features/step_defini
tions/link_steps.rb:83
    Then I can delete any user that I want to delete           # features/step_defini
tions/link_steps.rb:88
1 scenario (1 passed)
16 steps (16 passed)
```

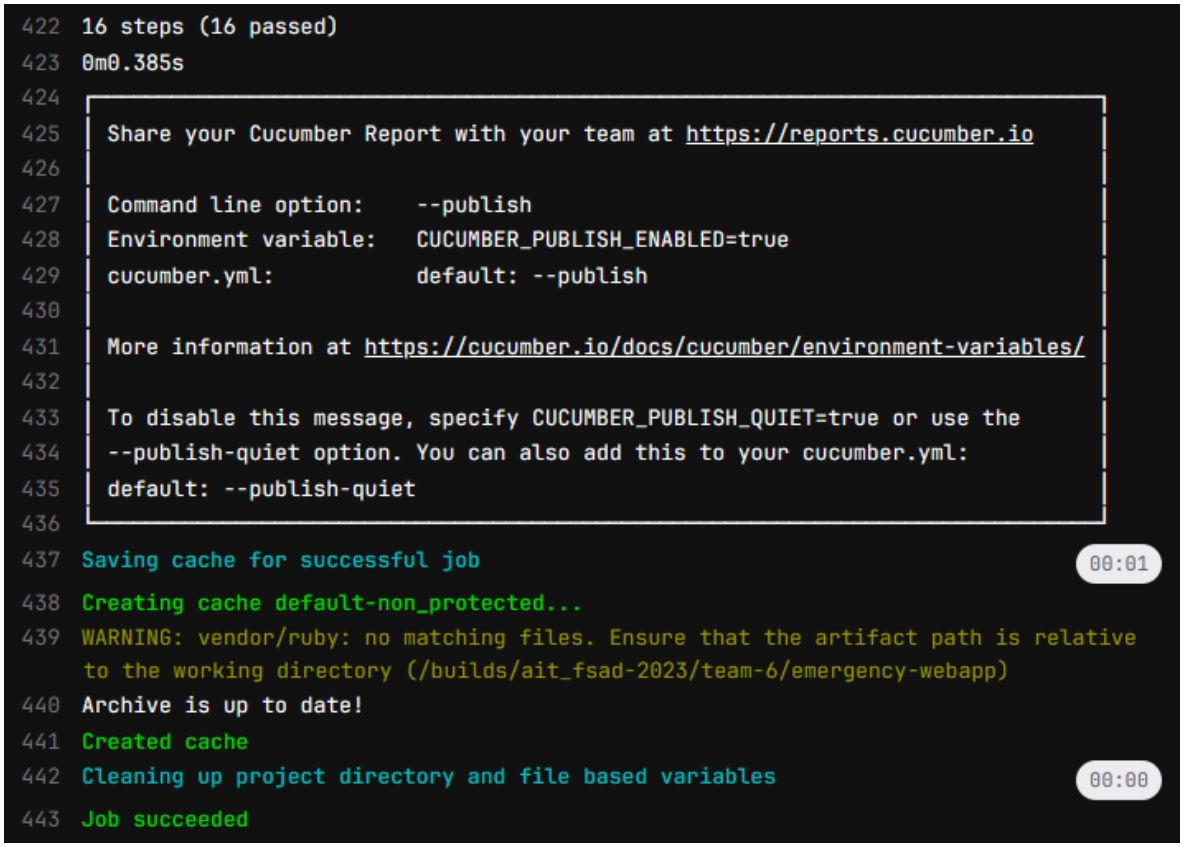


Figure 4.2 : some Cucumber test steps that has succeeded

4.3 Deployment We can able to deploy the website to the user that can access from real URL and make it to be automate deploy using Capistrano to the production. By the way, Deploying on EC2 is success but for the Capistrano, It has some errors occur for deployment. (Apache is not running : Some steps There are no Phusion Passenger(R)-served applications running whose paths begin) Although I can make sure that It can automate to create and migrate database.

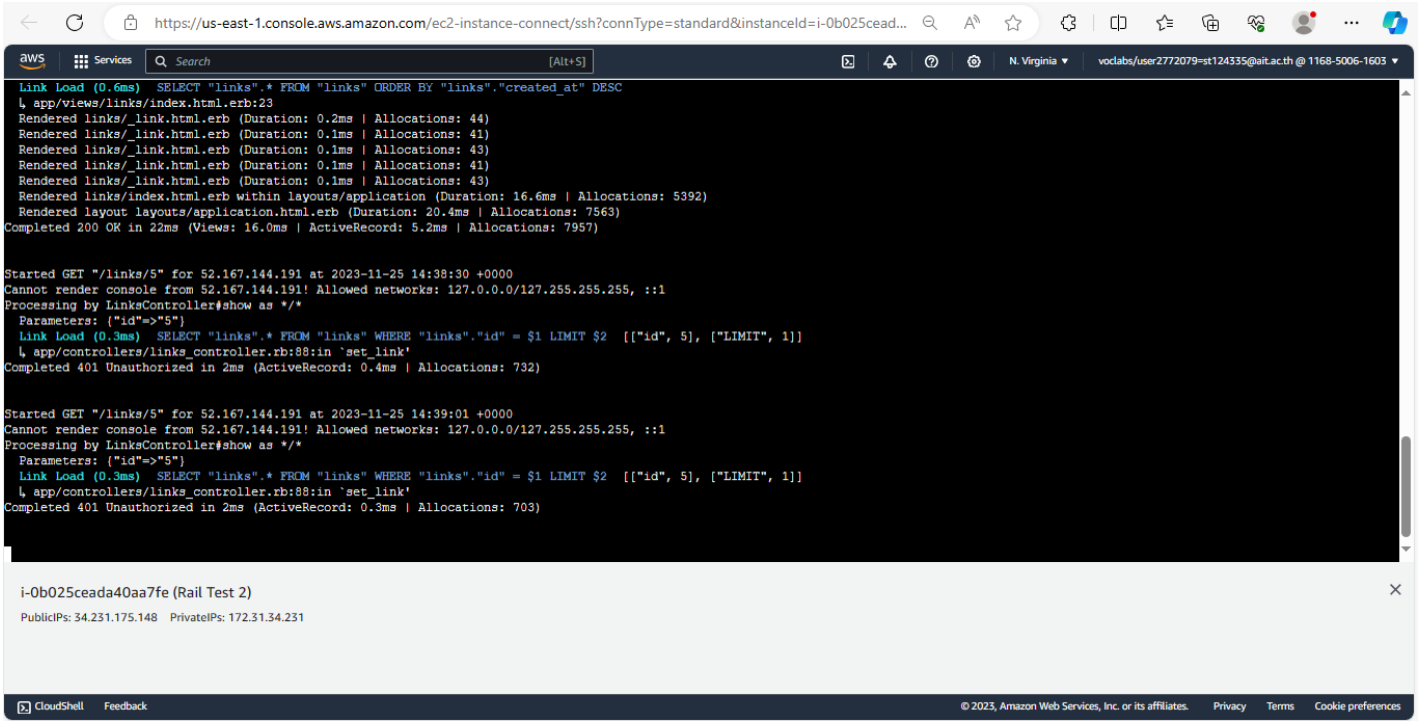


Figure 4.3.1 : Deployment on EC2 instance

```
nont18@nont18-VirtualBox:~/classprojec/readit$ cap production deploy BRANCH=main
00:00 git:wrapper
      01 mkdir -p /tmp
    ✔ 01 deploy@157.245.205.61 0.075s
      Uploading /tmp/git-ssh-f72fe06f2161493966b0.sh 100.0%
      02 chmod 700 /tmp/git-ssh-f72fe06f2161493966b0.sh
    ✔ 02 deploy@157.245.205.61 0.075s
00:00 git:check
      01 git ls-remote git@gitlab.com:ait_fsad-2023/team-6/emergency-webapp.git HEAD
      01 7ba1f65fda8353179f15c50f2abe5bd625f0baa5       HEAD
    ✔ 01 deploy@157.245.205.61 3.432s
00:03 deploy:check:directories
      01 mkdir -p /home/deploy/readit/shared /home/deploy/readit/releases
    ✔ 01 deploy@157.245.205.61 0.072s
00:03 deploy:check:linked_dirs
      01 mkdir -p /home/deploy/readit/shared/log /home/deploy/readit/shared/tmp/pids /home/deploy/readit/shared/tmp/cache /home/deploy/readit…
    ✔ 01 deploy@157.245.205.61 0.071s
00:04 git:clone
      The repository mirror is at /home/deploy/readit/repo
00:04 git:update
      01 git remote set-url origin git@gitlab.com:ait_fsad-2023/team-6/emergency-webapp.git
    ✔ 01 deploy@157.245.205.61 0.077s
      02 git remote update --prune
      02 Fetching origin
    ✔ 02 deploy@157.245.205.61 3.088s
00:07 git:create_release
      01 mkdir -p /home/deploy/readit/releases/20231125074544
    ✔ 01 deploy@157.245.205.61 0.072s
      02 git archive main | /usr/bin/env tar -x -f - -C /home/deploy/readit/releases/20231125074544
    ✔ 02 deploy@157.245.205.61 0.099s
00:07 deploy:set_current_revision
      01 echo "7ba1f65fda8353179f15c50f2abe5bd625f0baa5" > REVISION
    ✔ 01 deploy@157.245.205.61 0.090s

00:12 bundler:install
      The Gemfile's dependencies are satisfied, skipping installation
00:12 deploy:assets:precompile
      01 $HOME/.rbenv/bin/rbenv exec bundle exec rake assets:precompile
    ✔ 01 deploy@157.245.205.61 1.443s
00:13 deploy:assets:backup_manifest
      01 mkdir -p /home/deploy/readit/releases/20231125074544/assets_manifest_backup
    ✔ 01 deploy@157.245.205.61 0.072s
      02 cp /home/deploy/readit/releases/20231125074544/public/assets/.sprockets-manifest-62f11bdc7a410c4f895eb7f608a4e8ea.json /home/deploy/…
    ✔ 02 deploy@157.245.205.61 0.077s
00:13 deploy:migrate
      [deploy:migrate] Run `rake db:migrate`
00:14 deploy:migrating
      01 $HOME/.rbenv/bin/rbenv exec bundle exec rake db:migrate
    ✔ 01 deploy@157.245.205.61 1.364s
00:15 deploy:symlink:release
      01 ln -s /home/deploy/readit/releases/20231125074544 /home/deploy/readit/releases/current
    ✔ 01 deploy@157.245.205.61 0.075s
      02 mv /home/deploy/readit/releases/current /home/deploy/readit
    ✔ 02 deploy@157.245.205.61 0.074s
00:15 passenger:restart
      01 passenger-config restart-app /home/deploy/readit --ignore-app-not-running
      01 There are no Phusion Passenger(R)-served applications running whose paths begin with '/home/deploy/readit'.
    ✔ 01 deploy@157.245.205.61 0.204s
00:16 deploy:cleanup
      Keeping 5 of 6 deployed releases on 157.245.205.61
      01 rm -rf /home/deploy/readit/releases/20231123164658
    ✔ 01 deploy@157.245.205.61 0.085s
00:16 deploy:log_revision
      01 echo "Branch main (at 7ba1f65fda8353179f15c50f2abe5bd625f0baa5) deployed as release 20231125074544 by nont18" >> /home/deploy/readit…
    ✔ 01 deploy@157.245.205.61 0.072s
```

Figure 4.3.2 : Some steps that our team has deploy in Capistrano but Phusion Passenger does not work

5.Results

5.1Feature of application

5.1.1 User : User or student should have AIT student ID before enroll this system. This should include the domain name @ait.asia

- Each of User can post by writing some accidents to the community with Title, Descriptions, Type of emergency etc. when user already post to community, It will automatic show time that user already post

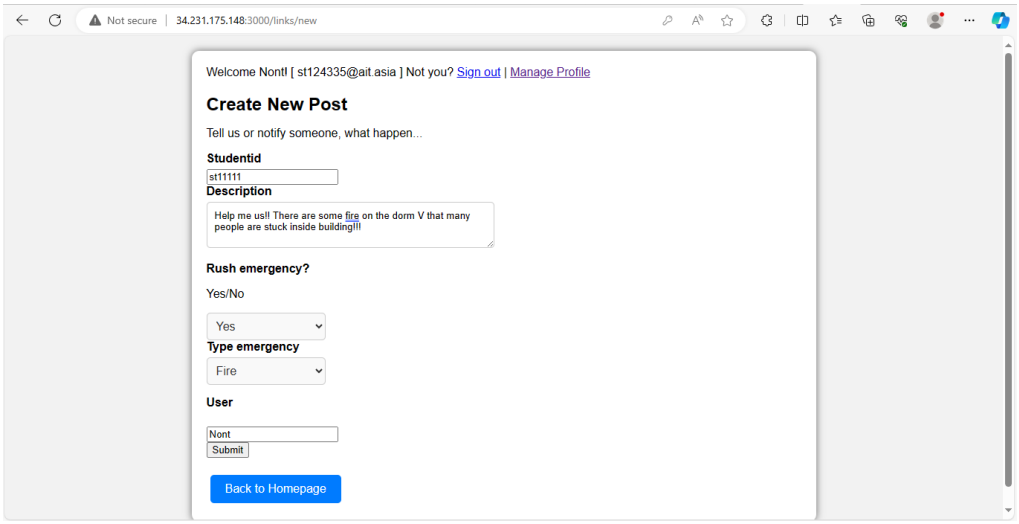


Figure 5.1.1.1 : User can post by writing some accidents to the community

- User who are the owner of the post can delete and update the post to the community for fixing by wrong typing

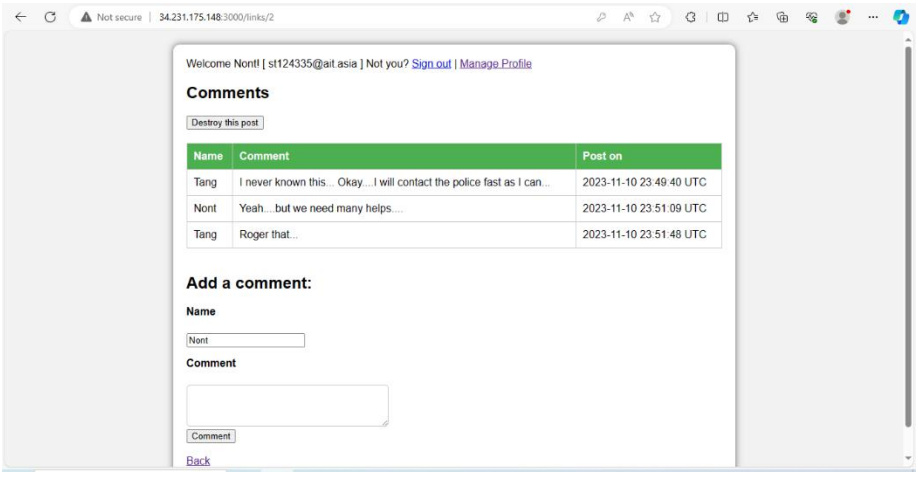


Figure 5.1.1.2 : User who are the owner of the post can delete and update post

- User can get some phone call see the map

- User can change password and cancel account anytime

5.1.2 Administrator : Administrator is the person who can see and manage some activities in website

- Administrator can delete all the post which they want to delete

- Administrator can create read update delete (CRUD) the phone list

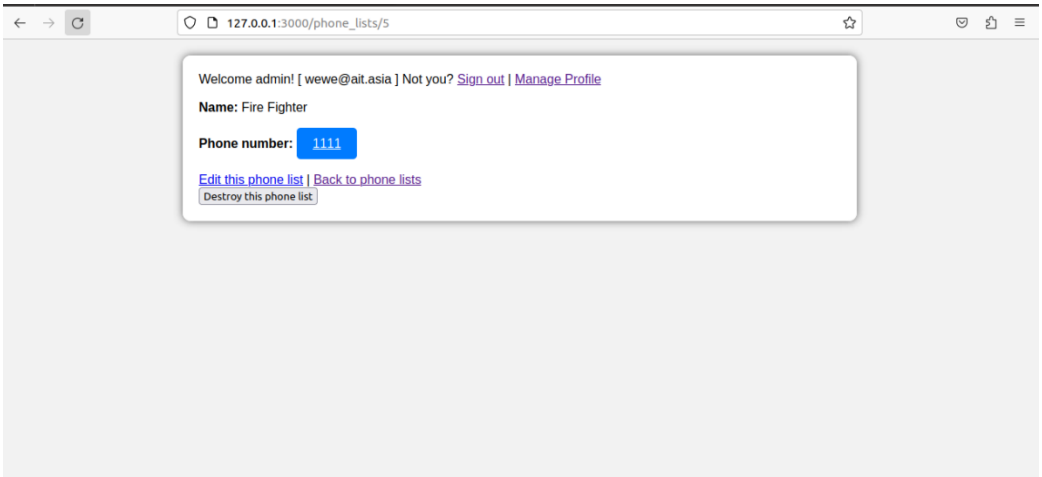


Figure 5.1.2.1 : Administrator can create read update delete (CRUD) the phone list

- Administrator can see the number of user and theirs name and the time when they had join to the system.

- Administrator can delete or kick user from the system

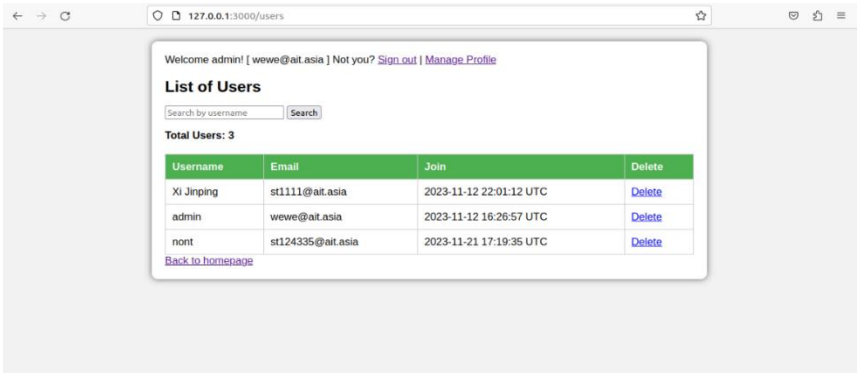


Figure 5.1.2.2 : Administrator can see the list of user and have a role to delete or kick user from the system

5.2 Reviews from some part of users

After deploying our website on EC2, we invited a segment of users to provide feedback. The feedback was diverse, featuring both positive and negative views on the web application. Below, We can outline some of the comments received as follows

User 1 : *"I think, for the user interface and functionality is great but It can attach some pictures or videos to imagine or see some result of accidents"*

User 2 : *"I like your idea, most website didn't create some kind like this. For the phone call page, It should be more access in hurry accident."*

User 3 : *"you don't have an anonymous submission feature. i believe it's important to protect the user's identity, especially when they want to make a report about something sensitive, or making report on behalf of someone else."*

By the way, All comments that we collect is valuable because user really want the feature that web-application can comfort them. We, as developer, should adapted and applied all those comments to the web-application.

5.3 Additional and Failure

For emergency website application, Our team is mostly done for all part such as role of user, user management for administration which can delete student account, manage some post in community or adding some data to the website. Although previously, We did not create and concern about user management page. We have developed the functionality and try to add Integration of maps that can auto detect user's location from smartphone or computer to migrate in the post. During developing can give us the longitudes and latitudes but we need to compare with the map. Finally, We need to cut this function off because It need to compare with the map and need to get specific each place.

```ruby
# db/seeds.rb
name = "Hôtel Martinez - The Unbound Collection by Hyatt"
address = "73 Bd de la Croisette, 06400 Cannes"
Location.create(name:, address:)

name = "Exclusive Hotel Belle Plage"
address = "2 Rue Brougham, 06400 Cannes"
Location.create(name:, address:)

name = "Best Western Premier Le Patio des Artistes - Cannes"
address = "6 Rue de Bône, 06400 Cannes"
Location.create(name:, address:)

name = "Le Negresco"
address = "37 Prom. des Anglais, 06000 Nice"
Location.create(name:, address:)

name = "Caesars Palace"
address = "3570 S Las Vegas Blvd, Las Vegas, NV 89109, United States"
Location.create(name:, address:)
```

```erb
# app/views/locations/show.html.erb
<% @location.nearbys(10).each do |location| %>
  <%= location.name %>
  <b>Distance:</b>
  <%= location.distance_to(@location).round(2) %>
  <%= Geocoder.config.units.to_s %>
  <br>
<% end %>
```

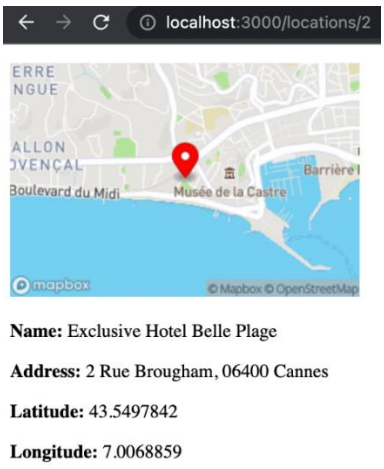


Figure 5.3 : prototype of code that can specify some of locations

Conclusions and future works

This website is all about emergency response within a community, built using Ruby on Rails (RoR), a model-view-controller (MVC) framework. It's a comprehensive platform that integrates various databases like PostgreSQL and presents views through html.erb, styled with CSS. Its primary function is to broadcast alerts or notifications regarding accidents or emergencies. While it might not be as quick as a phone call, it does feature a dedicated phone page with a list of SOS numbers for immediate emergencies.

The main aim here is to inform the community about incidents. Once students view posts and comments, they can spread the word, creating a chain reaction until most students are in the know. For the administrators, vital to the group, have full control using CRUD operations to manage the system effectively.

Looking ahead, the potential ward for this emergency website is vast. It could evolve into a more comprehensive tool, allowing users to upload images or videos, automatically detect user locations, and send real-time messages. This kind of integration would facilitate the sharing of vital information among people, fostering a stronger community bond and aiding in critical situations. We expect that this all features can develop and help some students in the forward feature.