VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

WEB PROGRAMMING (CO3049)

Assignment

BK-MEC: A Website for Finding Medical Doctor

CC01 - Semester 2022-2023

Instructor:	Mr. Nguyen Duc Thai	
Group Members:	To Nhat Duy	1952617
	Le Nguyen Gia Nghi	1952868
	Doan Viet Tu	1952521
	Trinh Minh Huy	1952730

Ho Chi Minh City, 11th December 2022

Contents

Document history

Date	Ver.	Changes	Author
10/12/2022	1.0	- Create report template, initialize sections.	Doan Viet Tu
10/12/2022	1.1	- Add introduction and requirements section.	Doan Viet Tu
10/12/2022	1.1	- Add design and development section.	To Nhat Duy
11/12/2022	1.2	- Adjust introduction section.	Trinh Minh Huy
11/12/2022	1.3	- Adjust development section.	Le Nguyen Gia Nghi
11/12/2022	1.4	- Adjust development section.	Trinh Minh Huy
11/12/2022	2.0	- Adjust design section.	To Nhat Duy
11/12/2022	2.1	- Adjust development section.	Le Nguyen Gia Nghi
11/12/2022	2.2	- Review and finish the report.	Doan Viet Tu

Work distribution

Stu. ID	Full name	Tasks	Finished
1952617	To Nhat Duy	- Main UI/back-end developer. - Implement database and system design.	100%
1952868	Le Nguyen Gia Nghi	- Main back-end developer. - Implement the development features.	100%
1952521	Doan Viet Tu	- Handle the report, support developing front-end. - Analyze the system requirements.	100%
1952730	Trinh Minh Huy	- Support developing front-end. - Summarize ideas and add slides.	100%

1 INTRODUCTION

1.1 About the project

The main objective of this project is to develop a simple but efficient website for **finding medical doctor** that can help patients immediately seek for medical health and book a doctor to their home. In this project, we have already completed almost every basic requirements as the course assignment requires.

Figure 1

1.2 Technologies used

1.2.1 MVC design pattern

In this project, we choose the MVC design pattern, in which MVC stands for **Model - View - Controller**, as the backbone architecture for the web application.

- **Model:**
  The Model component represents all the data-related logic that the user engages with. This can be the data being transmitted between the View and Controller components or any other data related to business logic.
- **View:**
  The View component is used to handle all of the application's UI logic.
- **Controller:**
  Controllers act as a link between the Model and View components, processing all business logic and incoming requests, manipulating data using the Model, and interacting with Views to render the final output.

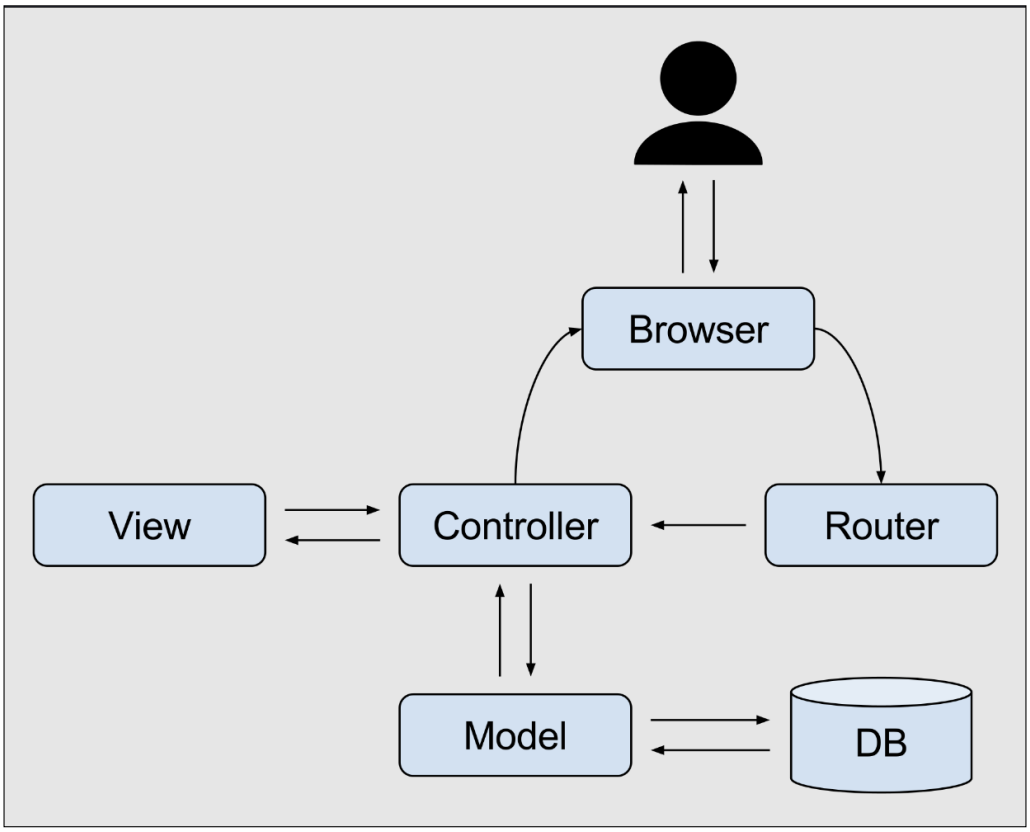

Figure 2

1.2.2 Framework

a. Django

This website is built mostly based on the Django framework. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development. Django is:

- Ridiculously fast: Django was designed to help developers take applications from concept to completion as quickly as possible.
- Reassuringly secure: Django takes security seriously and helps developers avoid many common security mistakes.
- Exceedingly scalable: Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

b. Bootstrap

This project also follows Bootstrap 4, which is a free front-end framework for faster and easier web development. We use this framework because of its responsive features, mobile-first approach and browser compatibility.

1.2.3 APIs

Google Maps API

Figure 3

Momo API

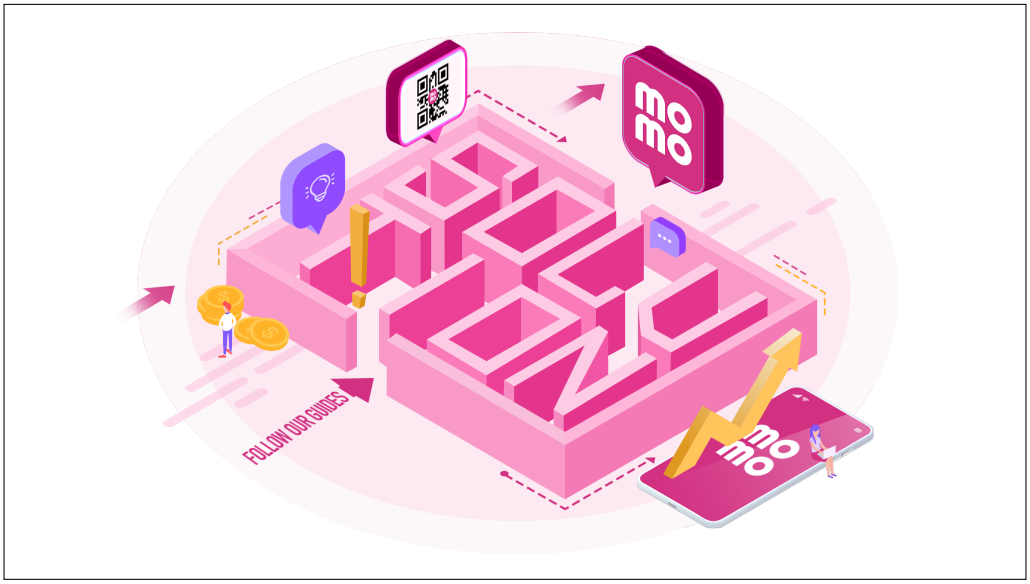

Figure 4

1.2.4 Libraries

There are some libraries used in the development process such as JQuery3.

2 REQUIREMENTS ANALYSIS

2.1 Functional requirements

- The web can display some advertisements or news banners.
- The web shall support a simple, categorized and well-looked list of doctors.
- Users can search for a doctor by their name.
- Users can click on each doctor to see the details.
- Users can login or signup to start booking.
- The booking form shall be integrated with Google Maps.
- Momo will be the priority payment method.
- The web can display on IOS/Android phones.

2.2 Non-functional requirements

- The web should have nice UI and easy-to-access functions.
- Users should easily find the right doctor and start booking in a few steps.
- The web is responsive and should load in less than 4 seconds.
- Only the system data administrator can assign roles and change access permissions to the system.
- The back-end complexity is minimized for future maintenance.

3 DESIGN

3.1 Database design

- Our group intends to develop an web application for booking doctor to come to patients's place and examine their health.
- In our design, we will have patients, doctor and admin distinguished by attribute 'role' of table users. A user can have many bookings and each booking will have only one doctor to take responsibility for.
- Moreover, to improve user experience and quality of the site when patients want to find a doctor who is an expert in a specialty. So, according to our design, we allow a doctor to have multiple specialties, and that will allow patients to filter easier.

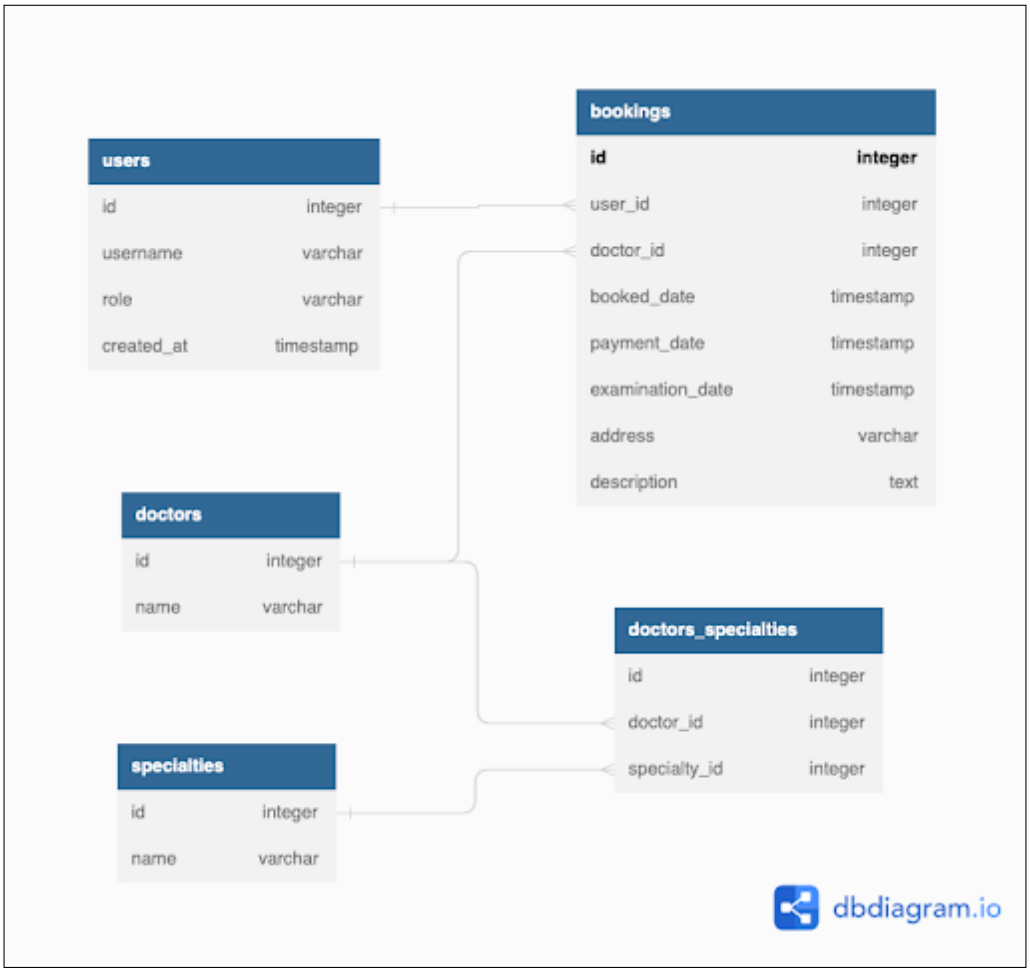

Figure 5

3.2 System design

- We end up with the monolithic design because the current system is quite small. Therefore database, web server, staticfiles, backend server all will be running on one single instance.
- We prefer to host on VPS, and use Ubuntu to be the operating system of the server because it's popular, well-supported, and the community is enormous. With the support of the community, we can easily find a solution when we get stuck. And, we love the apt package manager which allows us to install any required packages fast and simply.

- And, we use Apache2 to be a web server. The Apache HTTP Server is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Figure 6

4 DEVELOPMENT

4.1 Overview: Home page

Home page

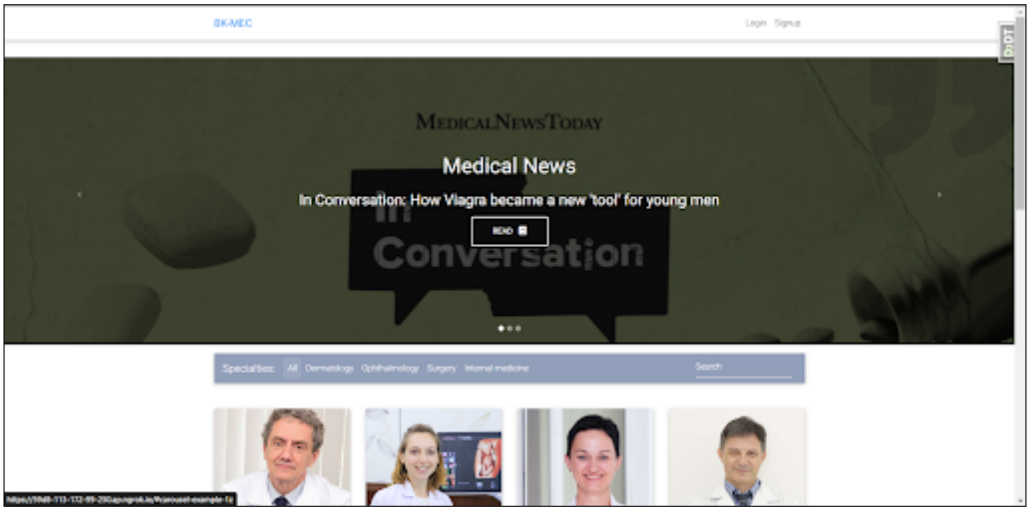

Figure 7

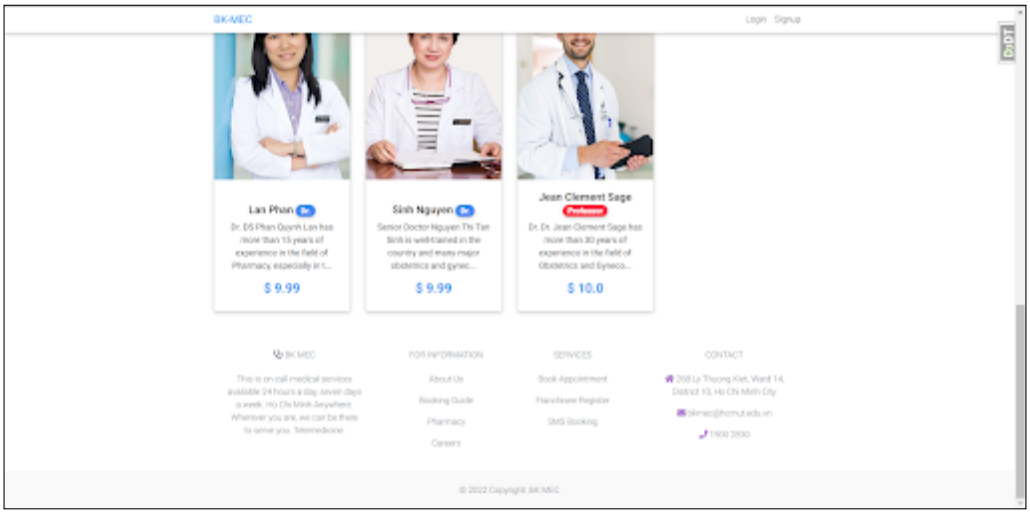

Figure 8

Firstly, Login and Signup buttons are located within the navigation bar on the top-right corner.

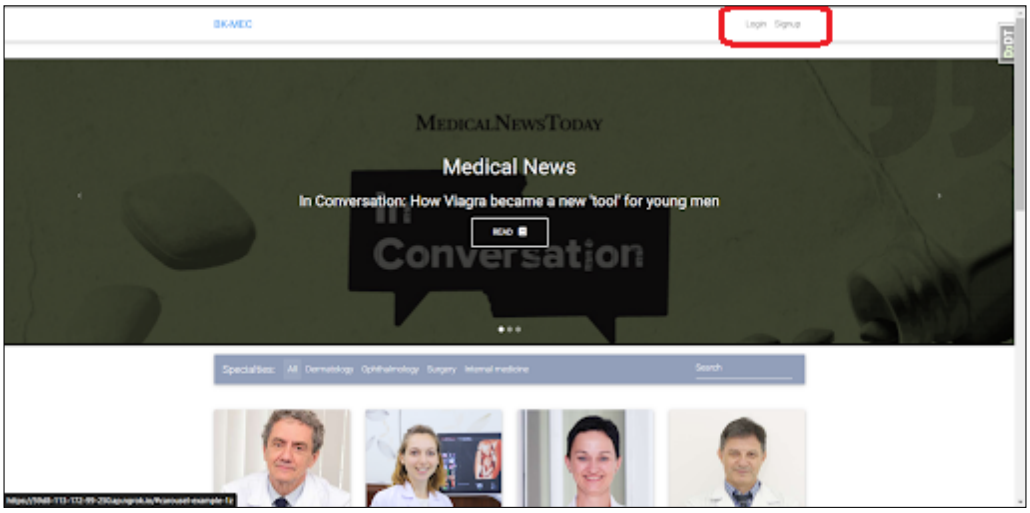

Figure 9

If already owning an account, users can easily login to the website.

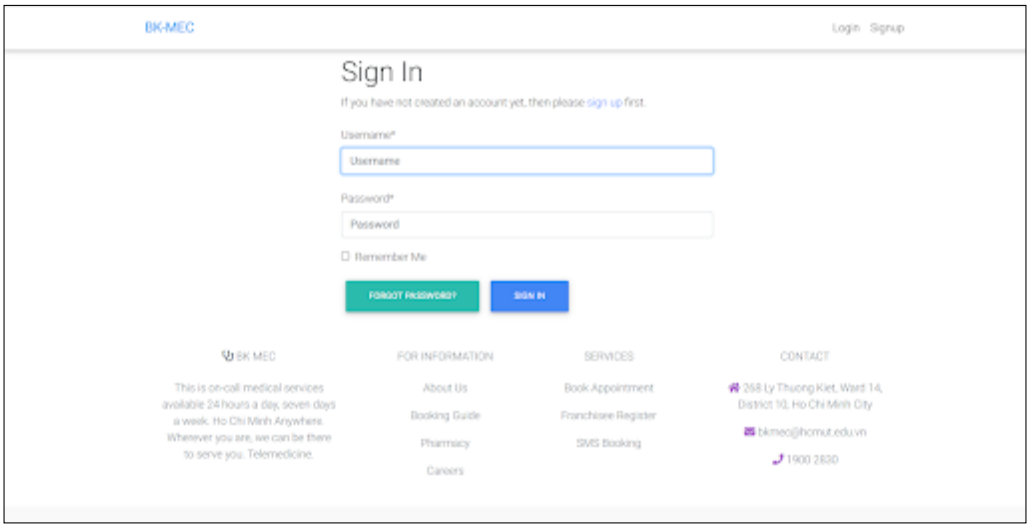

Figure 10

Otherwise, new users can create their account with the “Signup” button, which includes “username”, “password” and “email address”:

- “email address” must be in the right format.
- “password” must be at least 6 characters.

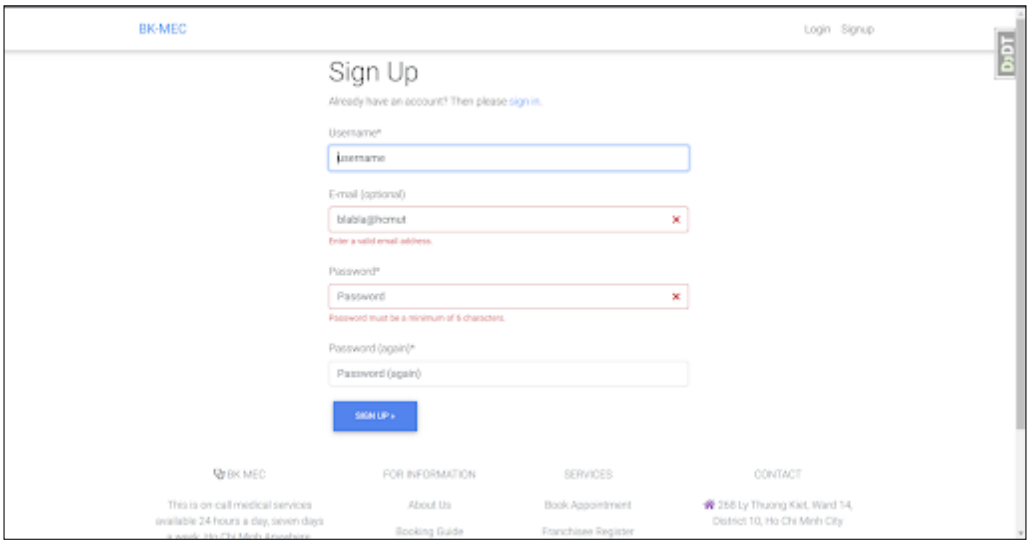

Figure 11

Back to home page

Figure 12

There are some faculties for customers to easily filter doctor who they need to find.

Figure 13

Also, search bar on the right to look for precisely the doctor users prefer.

Figure 14

If they want to learn more details about booking service, another services or misunderstanding can contact through the hotline.

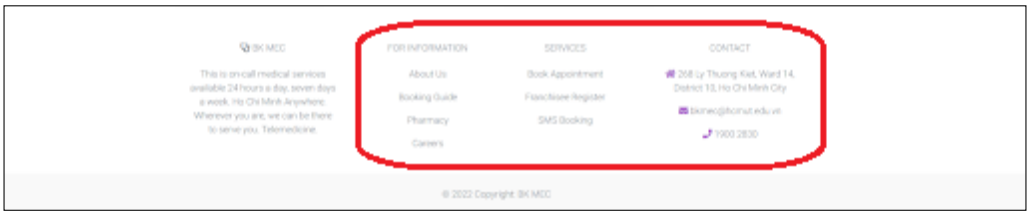

Figure 15

4.2 Doctors' details

Information for doctor about archives, achievements, researching places, universities,...

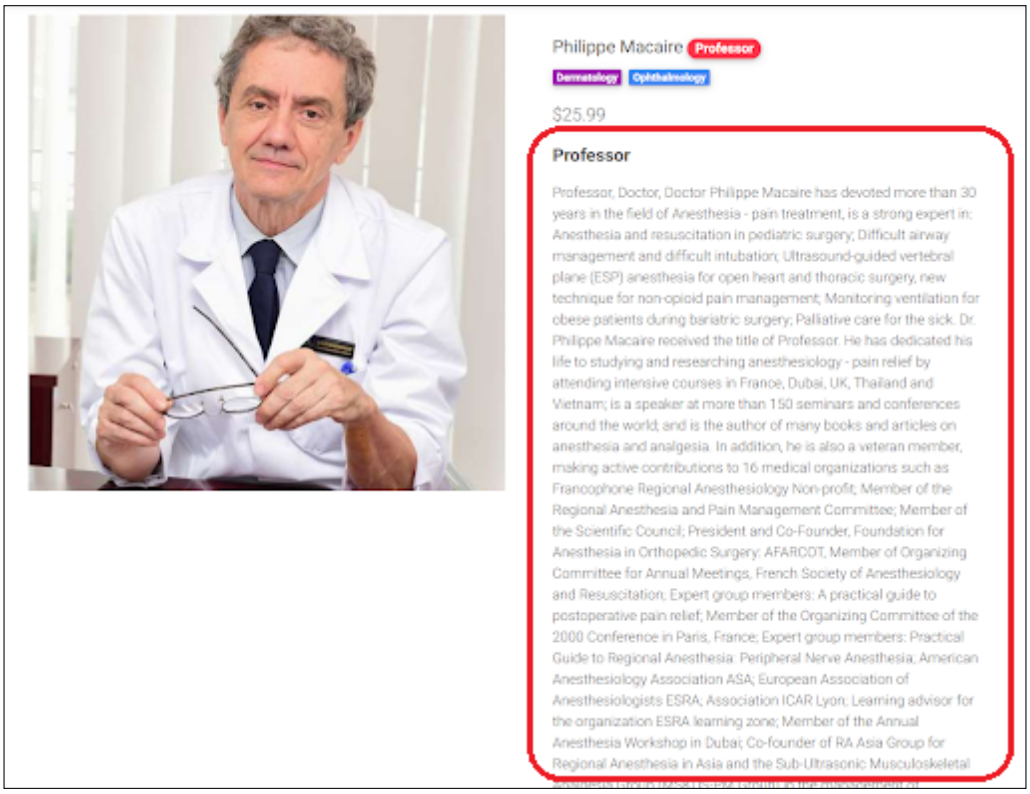

Figure 16

There are some briefs about name major, falcuties, price and status.

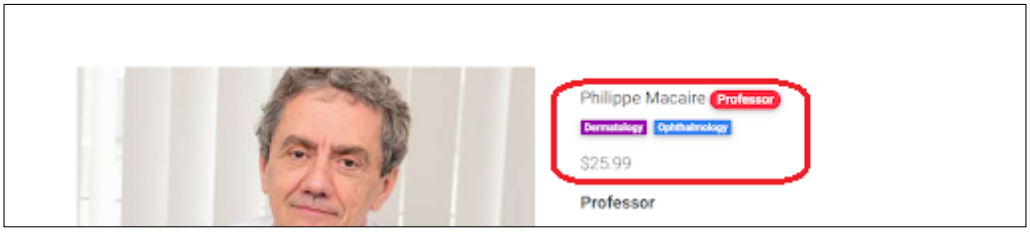

Figure 17

Next, the blue button used for booking doctor is lied below description of doctor and it will navigate to booking page.

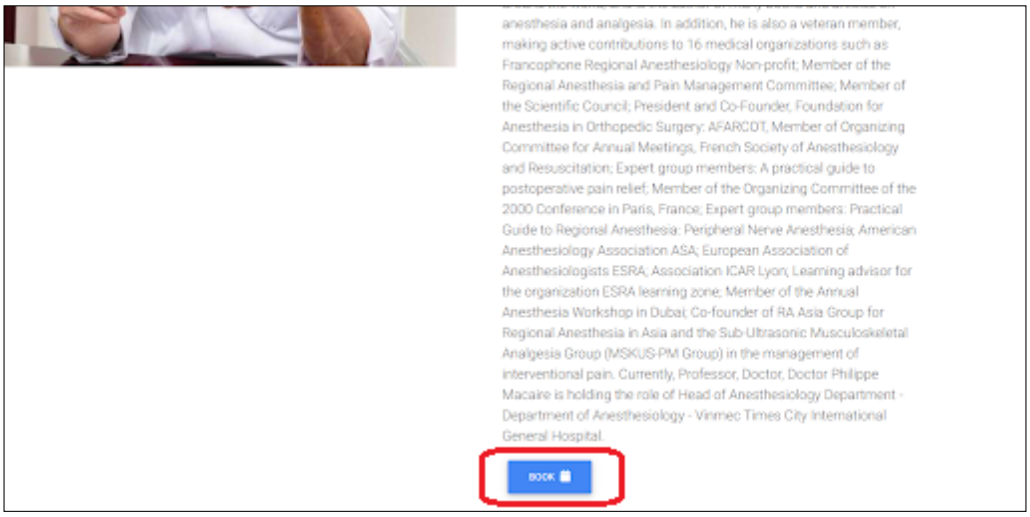

Figure 18

Last but not least, there is a recommendation section for clients to consider affordable price or the status of the others.

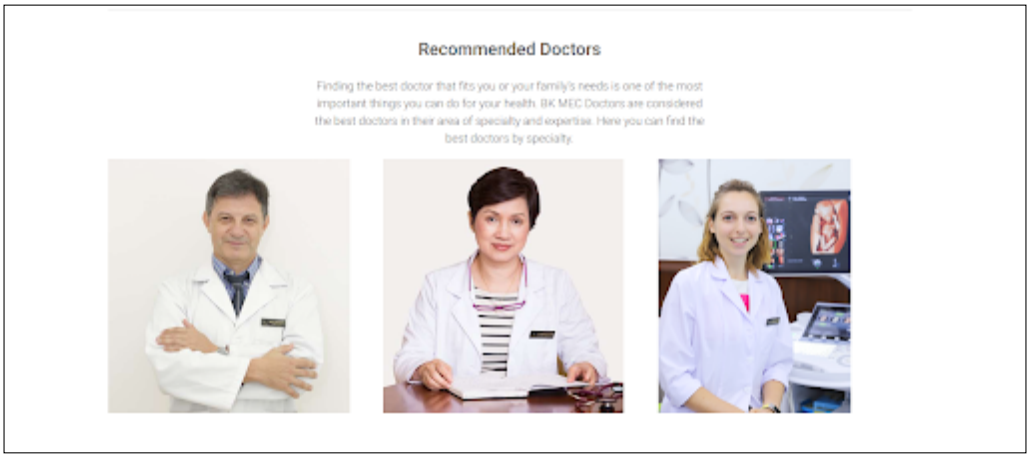

Figure 19

4.3 Booking form

Here we show some information about:

- Doctor info.
- Choose time to have an appointment.
- Show and select hospital with Google Maps.
- Describe your problem bar.
- Payment method by Momo.

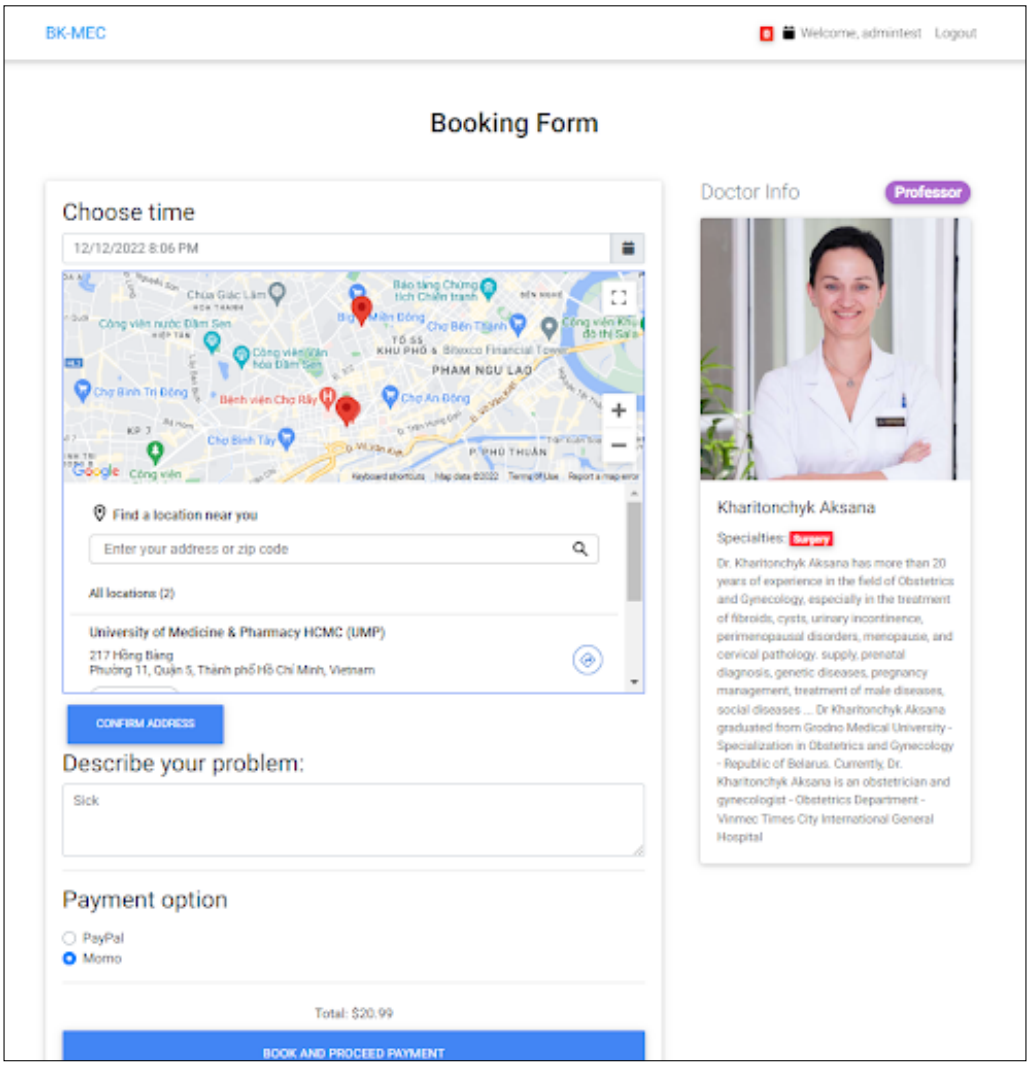
BK-MEC
Welcome, admintest
Logout
Booking Form
Choose time
12/12/2022 8:06 PM
Find a location near you
Enter your address or zip code
All locations (2)
University of Medicine & Pharmacy HCMC (UMP)
217 Hồng Bàng
Phường 11, Quận 5, Thành phố Hồ Chí Minh, Vietnam
CONFIRM ADDRESS
Describe your problem:
Sick
Payment option
PayPal
Momo
Total: $20.99
BOOK AND PROCEED PAYMENT
Doctor Info
Professor
Kharitonchyk Aksana
Specialties: Surgery
Dr. Kharitonchyk Aksana has more than 20 years of experience in the field of Obstetrics and Gynecology, especially in the treatment of fibroids, cysts, urinary incontinence, perimenopausal disorders, menopause, and cervical pathology. supply, prenatal diagnosis, genetic diseases, pregnancy management, treatment of male diseases, social diseases ... Dr Kharitonchyk Aksana graduated from Grodno Medical University - Specialization in Obstetrics and Gynecology - Republic of Belarus. Currently, Dr. Kharitonchyk Aksana is an obstetrician and gynecologist - Obstetrics Department - Vinmec Times City International General Hospital

Figure 20

4.3.1 Choose time

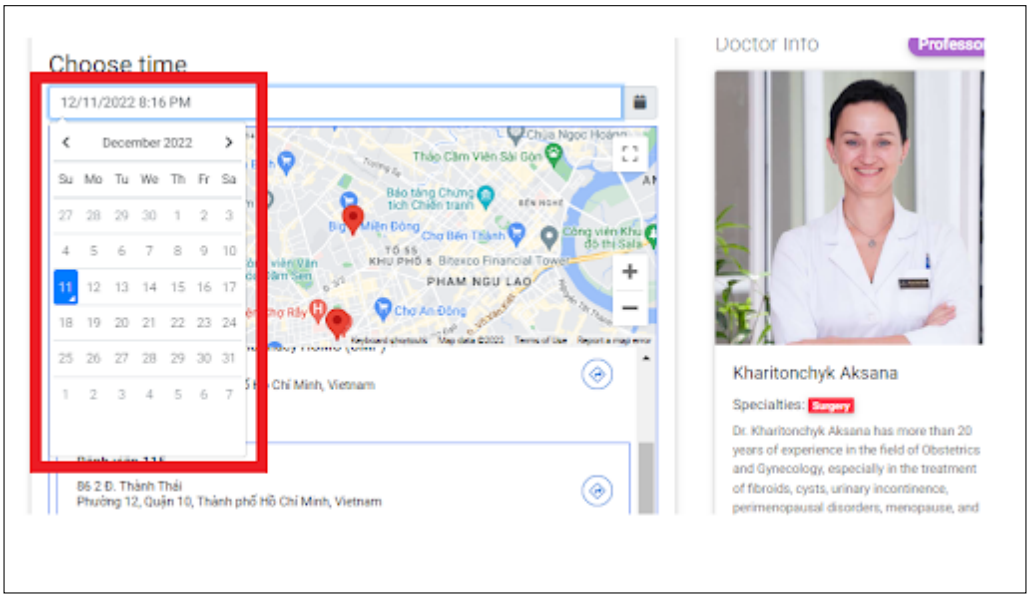

Figure 21

4.3.2 Choose location

About the location, we add 2 hospital so the patient can select which one they refer.

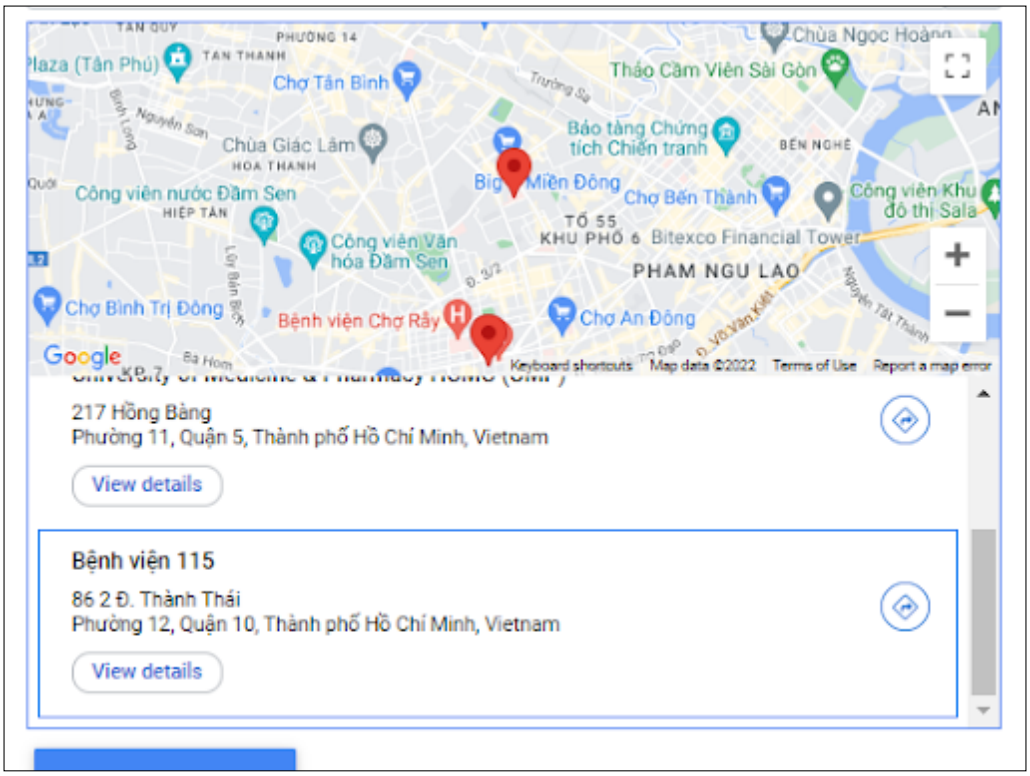

Figure 22

4.3.3 Describe the problem

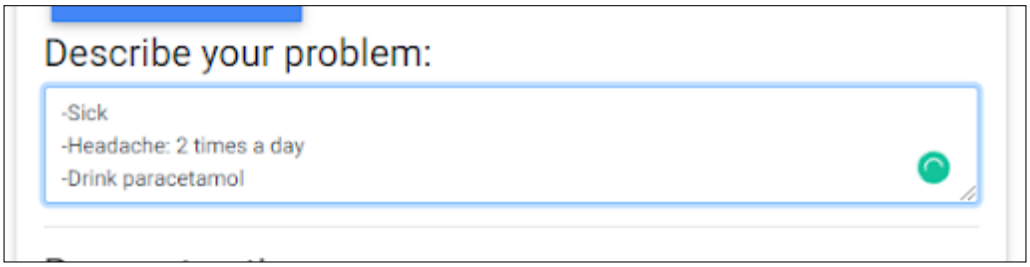

Figure 23

4.3.4 Confirm booking

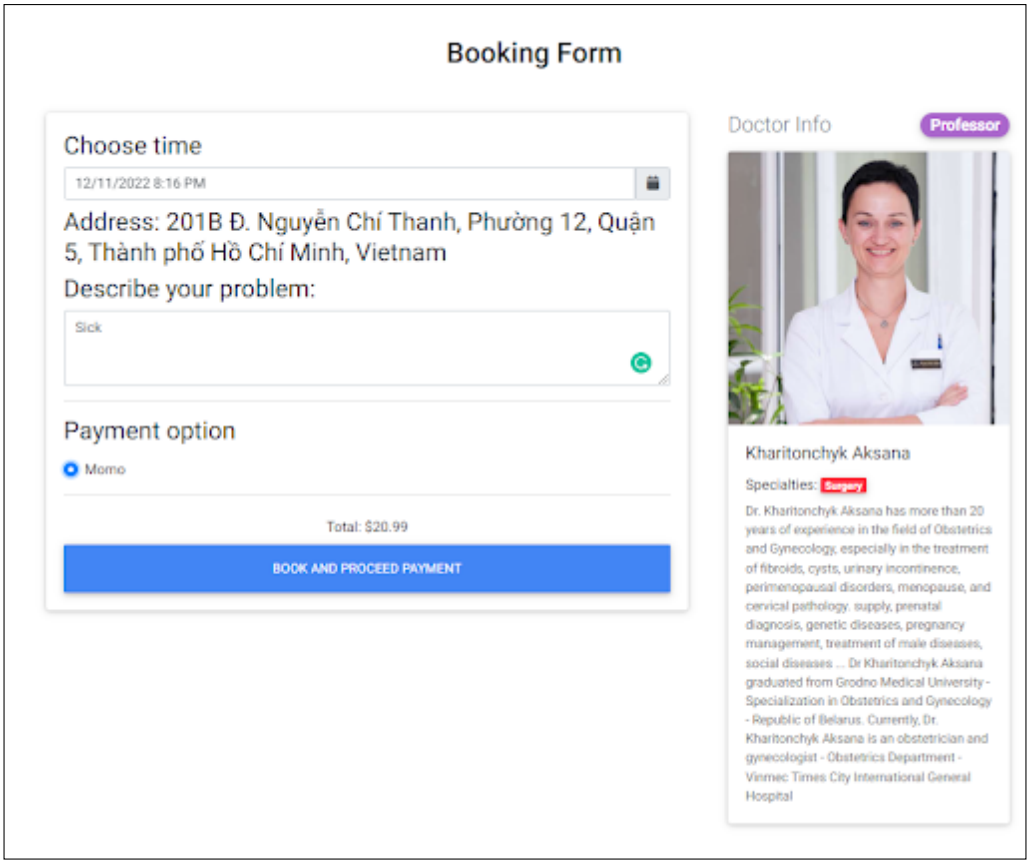

Figure 24

4.3.5 Payment

When we confirm payment, we collect all the information and send them to the MOMO Api, here we have already set up some basic information like endpoint, access key, secret key,... The Momo API just need the total amount of money, request id and then post it to the Momo Server, here we only use Momo API for developers and Momo Test only.

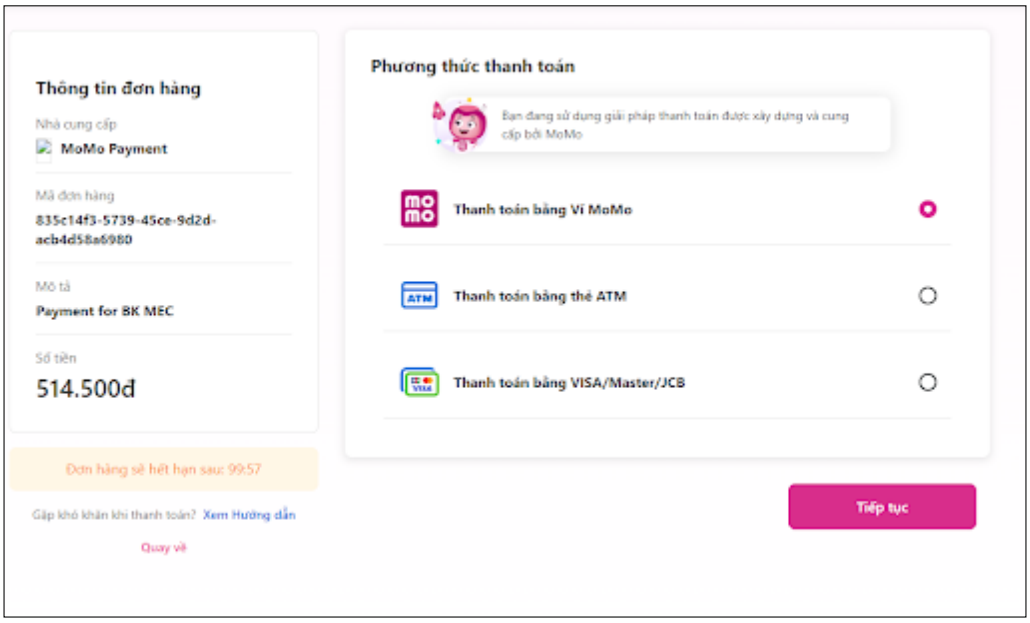

Figure 25

After sending the request, the transaction web pop up and the users can scan the QR code, if they don't have enough money, time out, or cancel the transaction, the web will return the home page.

```python
def momo_payment(amount, order_pk):
    # parameters send to MoMo get get payUrl
    endpoint = "https://test-payment.momo.vn/v2/gateway/api/create"
    accessKey = "F8BBA842ECF85"
    secretKey = "K951B6PE1waDMi640xX08PD3vg6EkVlz"
    orderInfo = "Payment for BK MEC"
    partnerCode = "MOMO"
    redirectUrl = f"http://127.0.0.1:8000/checkout/success/?pk={order_pk}"
    ipnUrl = "https://webhook.site/b3088a6a-2d17-4f8d-a383-71389a6c600b"
    amount = str(round(amount) * 24500)
    orderId = str(uuid.uuid4())
    requestId = str(uuid.uuid4())
    extraData = ""  # pass empty value or Encode base64 JsonString
    partnerName = "BK MEC"
    requestType = "payWithMethod"
    storeId = "BK"
    orderGroupId = ""
    autoCapture = True
    lang = "vi"
    orderGroupId = ""
```

Figure 26

4.4 Admin page

Django framework provides us an admin page, where we can do CRUD actions. From here, we can create/update/delete/read a model.

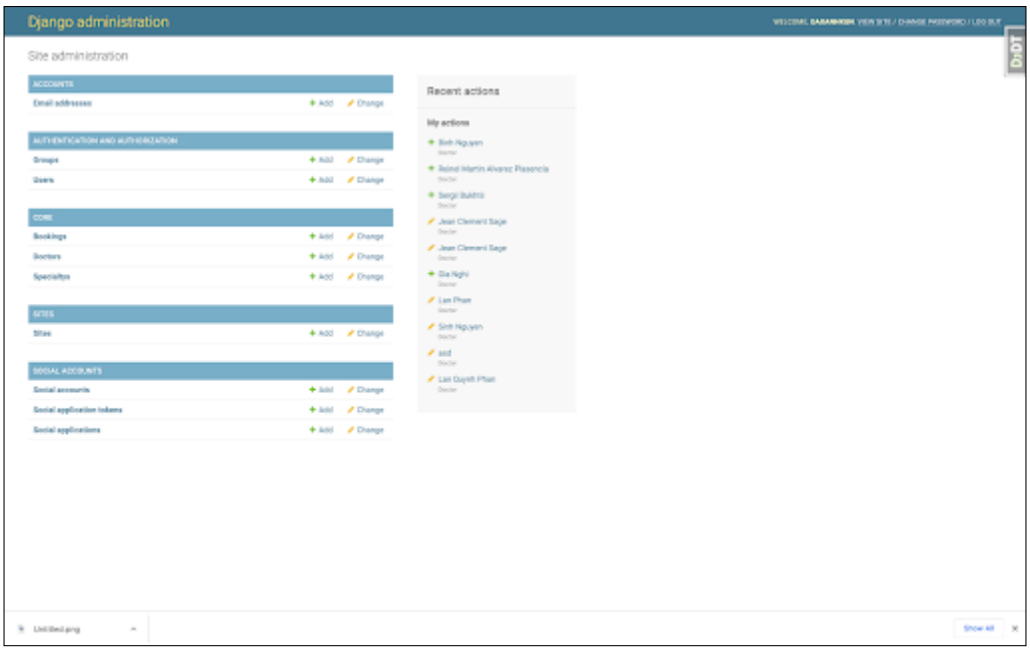

Figure 27

4.4.1 Add new data

Firstly, click the green add icon.

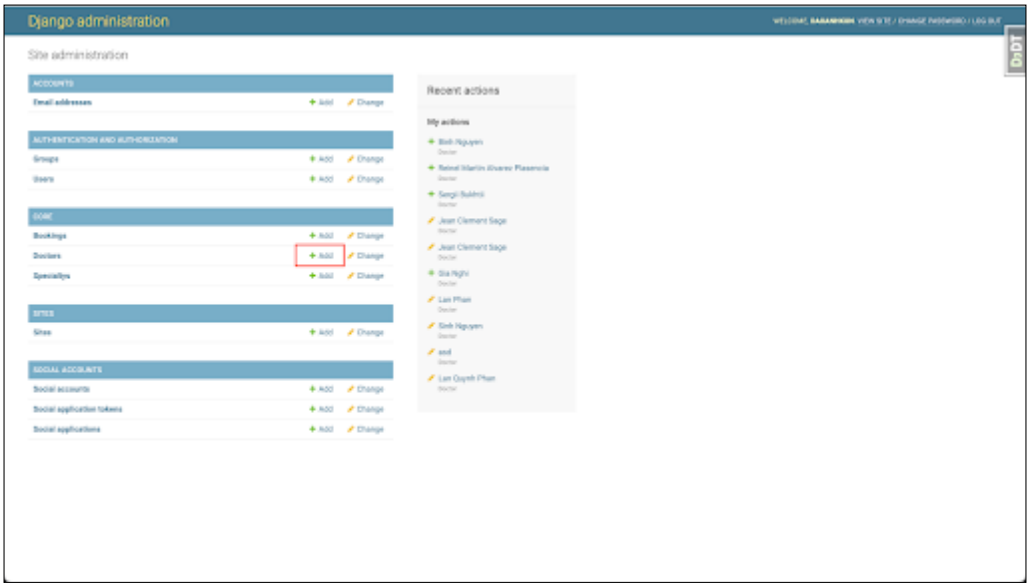

Figure 28

Then, fill up the form and click save.

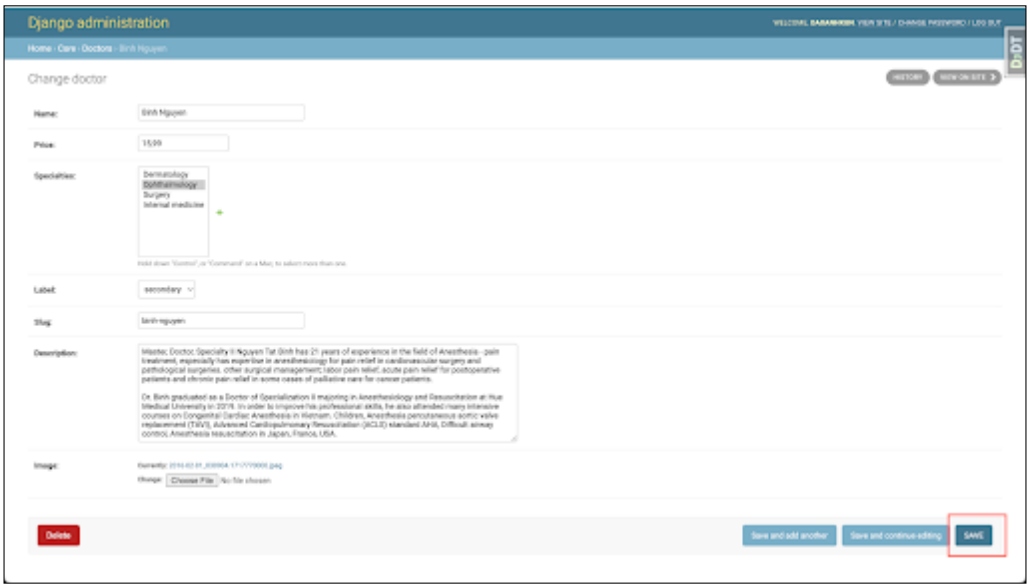

Figure 29

The doctor who we just added will be put into databases and we will see that doctor on homepage.

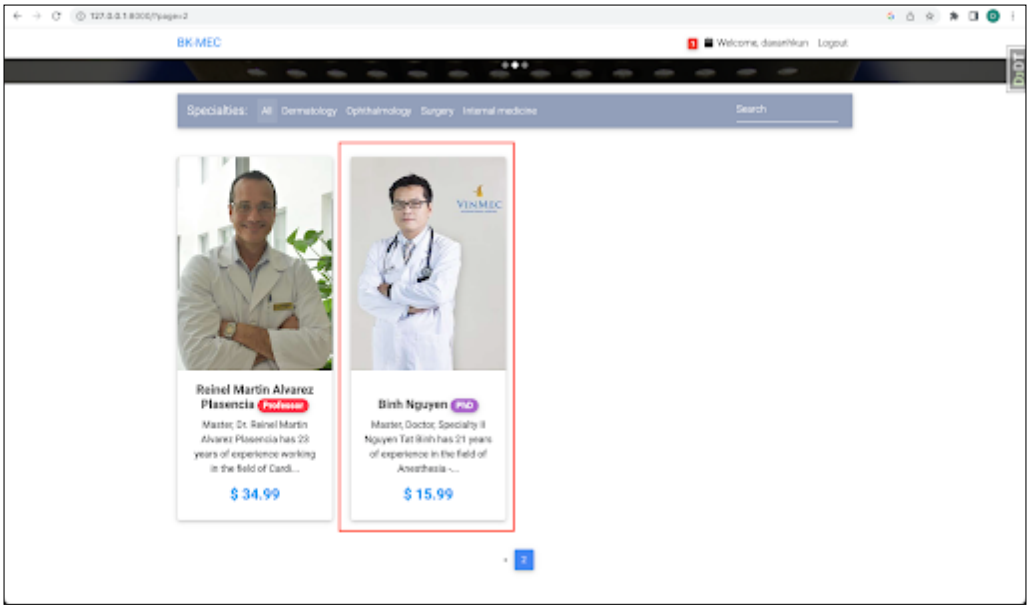

Figure 30

5 MAINTENANCE

5.1 Maintenance checklist

Website maintenance is the act of routinely checking the website for issues that negatively impact its performance and provide a poor user experience. These issues and tasks include:

- Broken links.
- Misspellings.
- Poor readability.
- Slow site speed.
- Security vulnerabilities.
- Content freshness.
- SEO errors.
- Conversion roadblocks.
- Backing up files.
- Updating software and plugins.

Scheduling in regular maintenance keeps the website operative, updated, and secure at all times. A healthy website means a better user experience, increased web traffic, and higher search engine rankings for your web pages. There are two ways to carry out website maintenance:

- Manually.
- Using an automated website maintenance service.

Regardless of the size and nature of the organization, website maintenance is considered as a business-critical task.

5.2 In this project

In this project, we plan to do some steps below to maintain the website for future use:

- Backing up files usually.
- Fixing HTML errors monthly.
- Developing new content if needed.
- Reviewing website analytics occasionally.

6 FINISHED PRODUCT

6.1 How to run

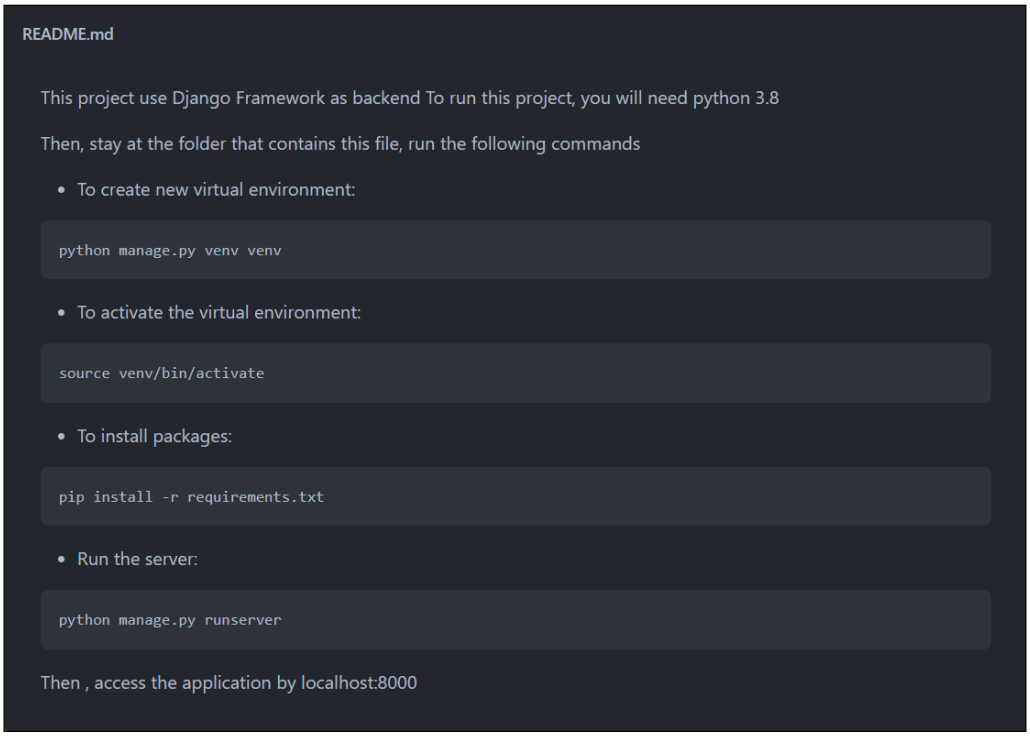

Figure 31

6.2 Source code

BK-MEC: Finding Medical Doctor | GitHub

References

[1] Adam D. Scott (2020). JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron. O'Reilly Media.