Object Oriented Programming

PROJECT REPORT

Course Code: CSE 234

Submitted to:

Md. Zahid Hasan

Associate Professor

Daffodil International University

Date of Submission:
12/5/2022

Team Member

1. Samsun Naher Asme ID no: 203-15-3882 Section: PC-A Department of CSE	**2. Md. Rakibul Islam Shanto** ID no: 203-15-3871 Section: PC-A Department of CSE

Project Name

Airline Management System

Abstract

We have a keen interest in aircraft. That is why we made an effort to address the airplane. Traveling by airline is the fastest option. Thus, it is imperative. There are numerous transportation options, including buses and trains. This one is like all the others. We attempted to start with this as our first Django project.

Introduction

Nowadays, any company needs have a website to display their products and services, and this project also represents a company that operates in the airline industry. We utilize Django, HTML, VS Code, Anaconda, and Xampp to create this kind of website. Python-based Django is a very well-liked and feature-rich server-side web framework. This project demonstrates the benefits of using Django, one of the most well-liked web server frameworks, to build your own websites. This website is very user-friendly and straightforward to navigate. Although it doesn't have many functions, it does contain those that are essential for an aviation management system.

Method

We have studied how Django functions after learning about the fundamentals of the framework and other components required to create any type of website.

Basically, a web application waits for HTTP requests from the web browser on a standard data-driven website. The program determines what is required in response to a request based on the URL and sometimes data from a POST or GET request. It may then read or write data from a database or carry out additional operations to fulfill the request, depending on what is necessary. The program will then reply to the web browser by placing the acquired data into placeholders in an HTML template, frequently producing an HTML page dynamically for the browser to view. Knowing about Django, I used Anaconda PowerShell to create an environment. I started working on my frontend, or the design portion, using VS Code to build, html to create the pages, and Bootstrap to design.

Django

A high-level Python web framework called Django enables the quick creation of safe and dependable websites. Django, which was created by seasoned programmers, handles a lot of the hassle associated with web development, allowing you to concentrate on developing your app without having to invent the wheel.

Templates: A template is a text file that specifies the organization or format of a file. Actual content is represented by placeholders. Using an HTML template and data from a model, a view can dynamically produce an HTML page. Any type of file can have its structure defined by a template; HTML is not required!

View: A view is a function that receives HTTP requests and responds to those requests with HTTP. Models provide views with access to the data they require to fulfill requests, and templates are given control over the formatting of the response.

Models: Models are Python objects that specify the data structure of an application and offer tools for managing (adding, deleting, and querying) database entries.

Urls: Writing distinct view functions to handle each resource is significantly more maintainable than processing requests from all URLs at once using a single function. Based on the request URL, a URL mapper is utilized to direct HTTP requests to the relevant view. The URL mapper can also look for specific strings or number patterns in a URL and pass those patterns as data to a view function.

Background

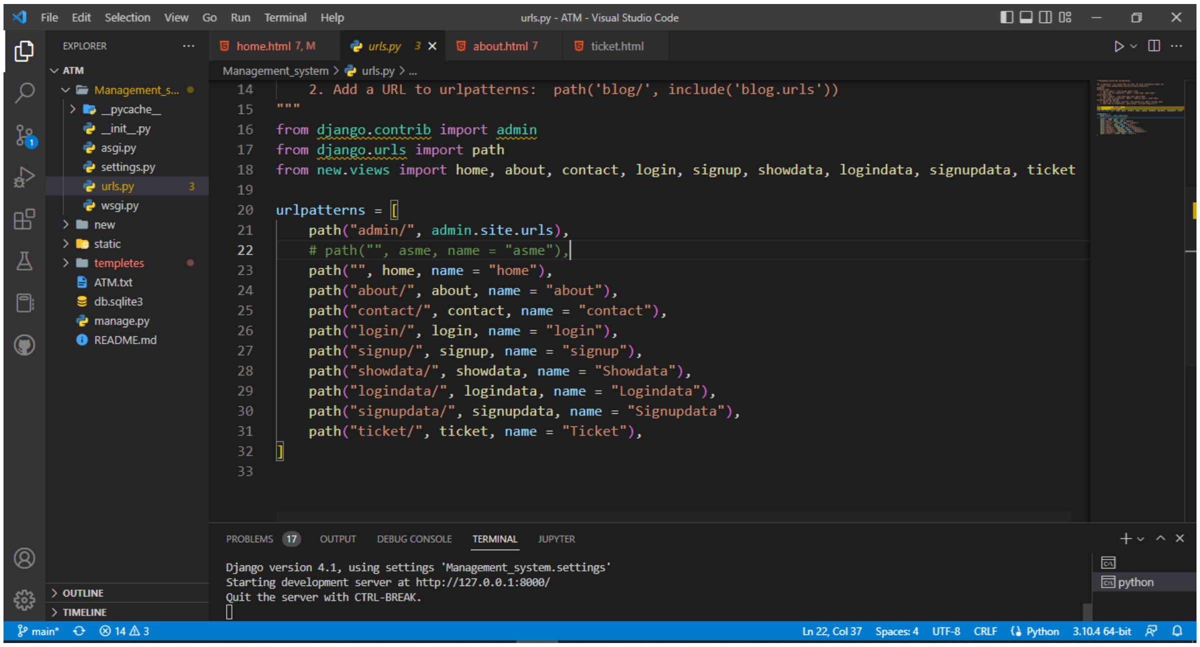
This project was created utilizing software like VSCode, Annaconda, Utilize a database, Create a table, add an image, use HTML and CSS, utilizing Python, HTML, Django, and Bootstrap.

Backend

A database, a server, and an application make up a website's back end. The technology that drives the many components that collectively allow the user-facing portion of the website to even exist is created and maintained by a back-end developer.

Therefore, if we disassembled our backend, there would be a projects file, project's app files, static files including css files, html files, which are my webpages, and finally a requirement page that included the project's needs.

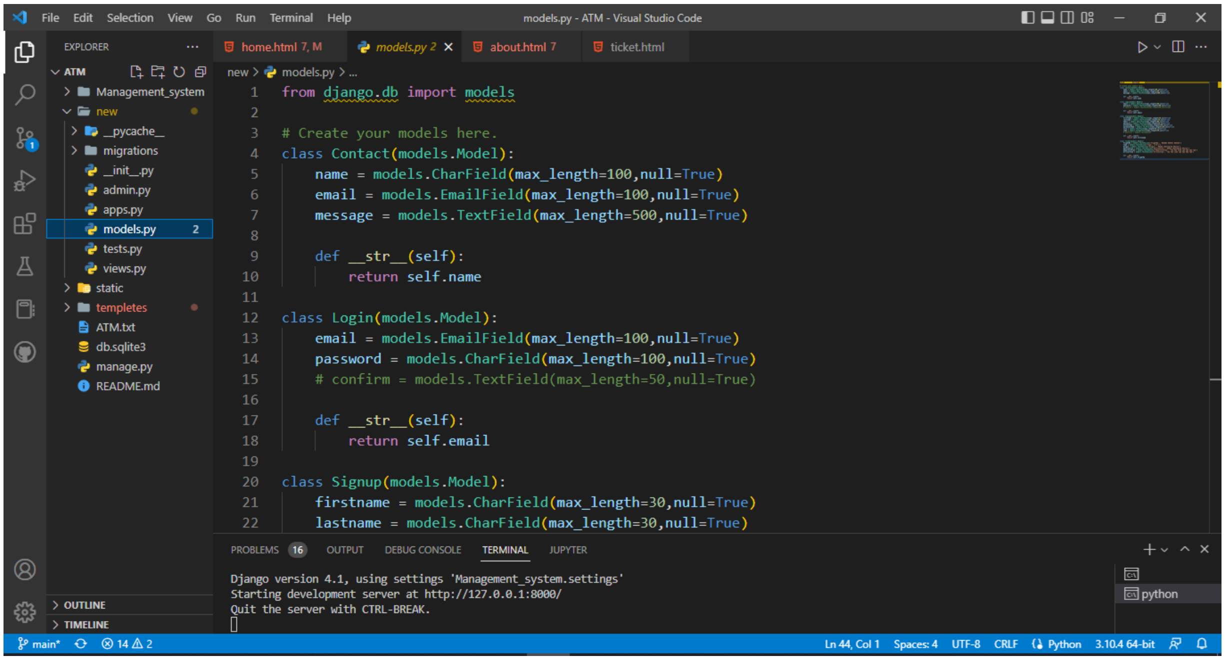
Urls.py



```
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from new.views import home, about, contact, login, signup, showdata, logindata, signupdata, ticket

urlpatterns = [
    path("admin/", admin.site.urls),
    # path("", asme, name = "asme"),
    path("", home, name = "home"),
    path("about/", about, name = "about"),
    path("contact/", contact, name = "contact"),
    path("login/", login, name = "login"),
    path("signup/", signup, name = "signup"),
    path("showdata/", showdata, name = "Showdata"),
    path("logindata/", logindata, name = "Logindata"),
    path("signupdata/", signupdata, name = "Signupdata"),
    path("ticket/", ticket, name = "Ticket"),
]

```

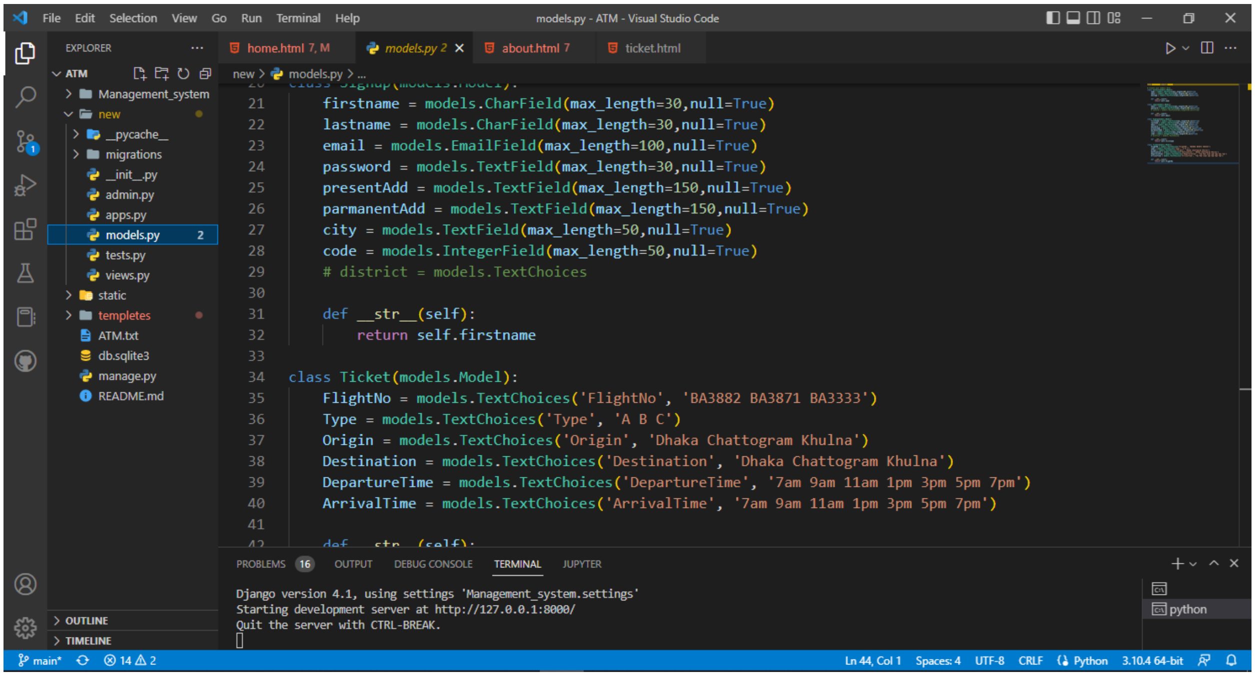
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Models.py



```python
from django.db import models

# Create your models here.
class Contact(models.Model):
    name = models.CharField(max_length=100,null=True)
    email = models.EmailField(max_length=100,null=True)
    message = models.TextField(max_length=500,null=True)

    def __str__(self):
        return self.name

class Login(models.Model):
    email = models.EmailField(max_length=100,null=True)
    password = models.CharField(max_length=100,null=True)
    # confirm = models.TextField(max_length=50,null=True)

    def __str__(self):
        return self.email

class Signup(models.Model):
    firstname = models.CharField(max_length=30,null=True)
    lastname = models.CharField(max_length=30,null=True)
```

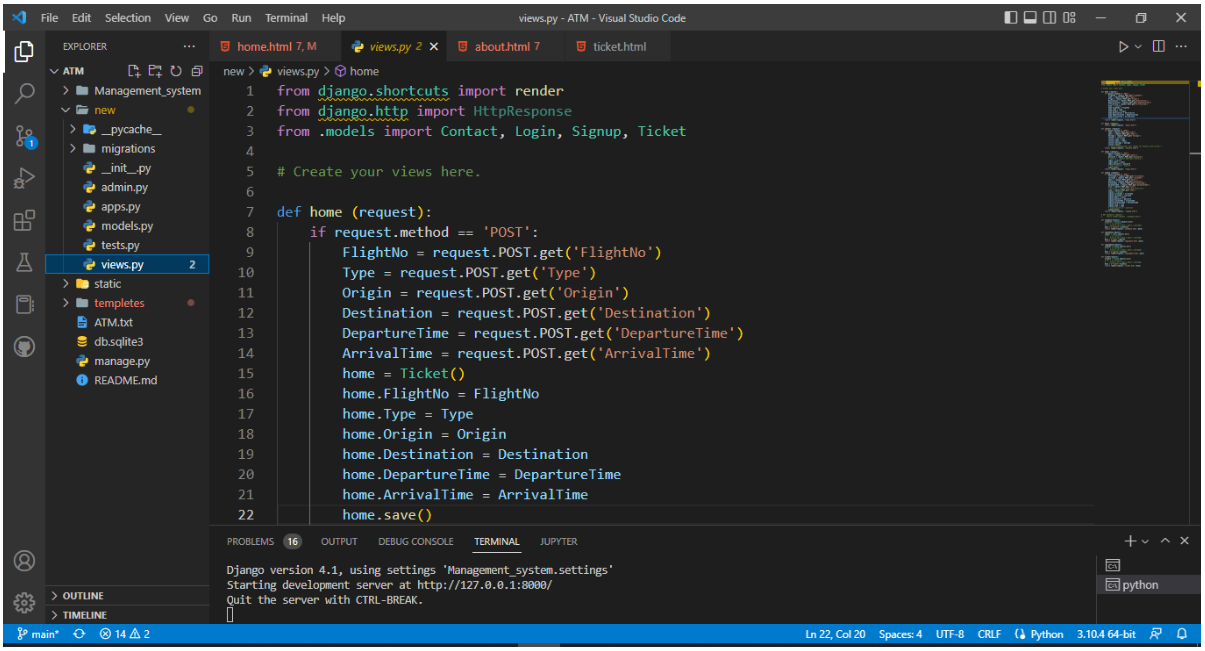
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```python
class Signup(models.Model):
    firstname = models.CharField(max_length=30,null=True)
    lastname = models.CharField(max_length=30,null=True)
    email = models.EmailField(max_length=100,null=True)
    password = models.TextField(max_length=30,null=True)
    presentAdd = models.TextField(max_length=150,null=True)
    parmanentAdd = models.TextField(max_length=150,null=True)
    city = models.TextField(max_length=50,null=True)
    code = models.IntegerField(max_length=50,null=True)
    # district = models.TextChoices

    def __str__(self):
        return self.firstname

class Ticket(models.Model):
    FlightNo = models.TextChoices('FlightNo', 'BA3882 BA3871 BA3333')
    Type = models.TextChoices('Type', 'A B C')
    Origin = models.TextChoices('Origin', 'Dhaka Chattogram Khulna')
    Destination = models.TextChoices('Destination', 'Dhaka Chattogram Khulna')
    DepartureTime = models.TextChoices('DepartureTime', '7am 9am 11am 1pm 3pm 5pm 7pm')
    ArrivalTime = models.TextChoices('ArrivalTime', '7am 9am 11am 1pm 3pm 5pm 7pm')

    def __str__(self):
```

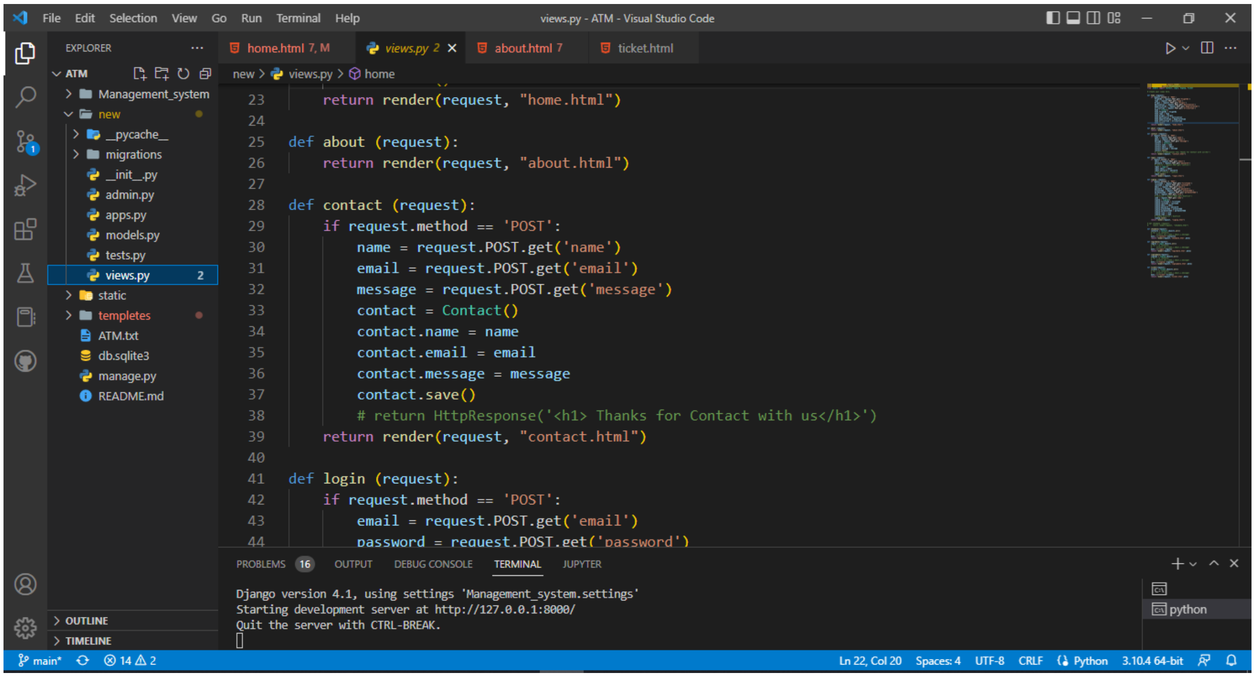
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Views.py



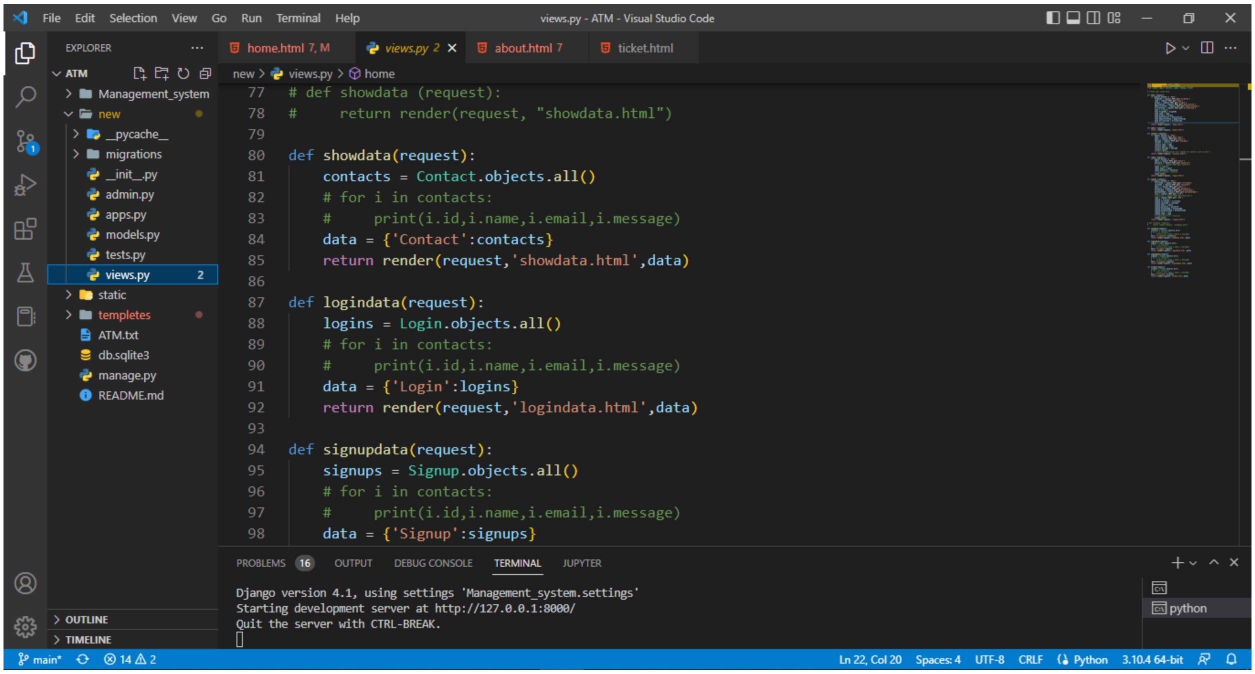
```python
from django.shortcuts import render
from django.http import HttpResponse
from .models import Contact, Login, Signup, Ticket

# Create your views here.

def home (request):
    if request.method == 'POST':
        FlightNo = request.POST.get('FlightNo')
        Type = request.POST.get('Type')
        Origin = request.POST.get('Origin')
        Destination = request.POST.get('Destination')
        DepartureTime = request.POST.get('DepartureTime')
        ArrivalTime = request.POST.get('ArrivalTime')
        home = Ticket()
        home.FlightNo = FlightNo
        home.Type = Type
        home.Origin = Origin
        home.Destination = Destination
        home.DepartureTime = DepartureTime
        home.ArrivalTime = ArrivalTime
        home.save()
```

```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```python
    return render(request, "home.html")

def about (request):
    return render(request, "about.html")

def contact (request):
    if request.method == 'POST':
        name = request.POST.get('name')
        email = request.POST.get('email')
        message = request.POST.get('message')
        contact = Contact()
        contact.name = name
        contact.email = email
        contact.message = message
        contact.save()
        # return HttpResponse('<h1> Thanks for Contact with us</h1>')
    return render(request, "contact.html")

def login (request):
    if request.method == 'POST':
        email = request.POST.get('email')
        password = request.POST.get('password')
```

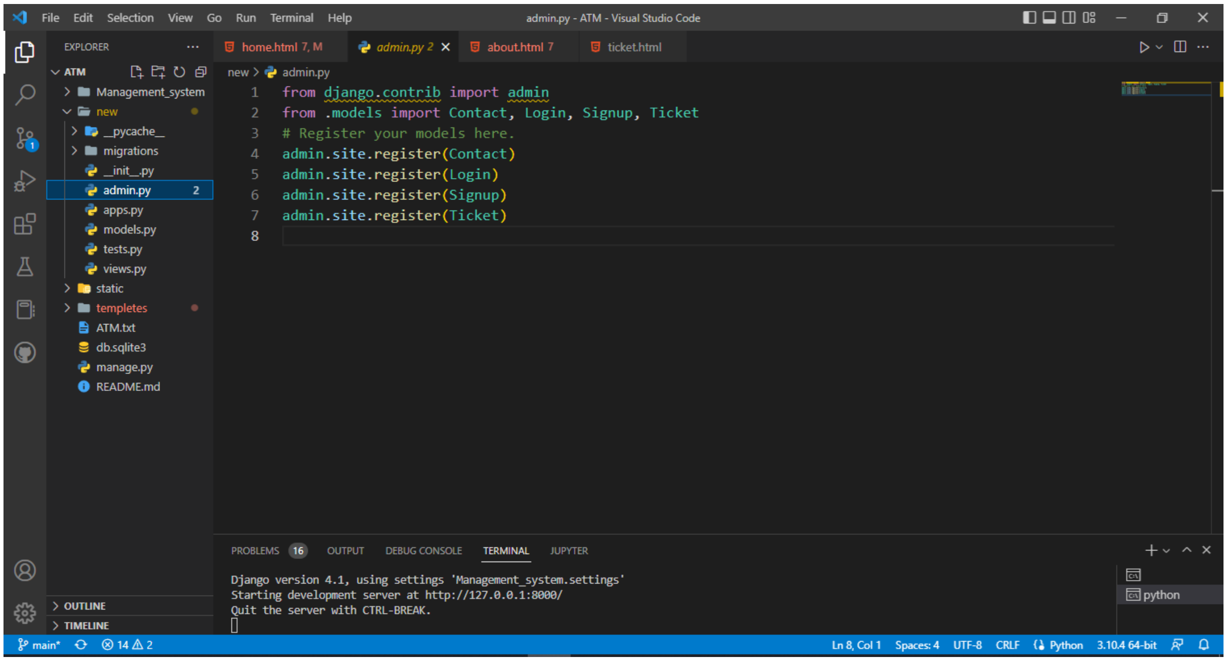
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```python
# def showdata (request):
#     return render(request, "showdata.html")

def showdata(request):
    contacts = Contact.objects.all()
    # for i in contacts:
    #     print(i.id,i.name,i.email,i.message)
    data = {'Contact':contacts}
    return render(request,'showdata.html',data)

def logindata(request):
    logins = Login.objects.all()
    # for i in contacts:
    #     print(i.id,i.name,i.email,i.message)
    data = {'Login':logins}
    return render(request,'logindata.html',data)

def signupdata(request):
    signups = Signup.objects.all()
    # for i in contacts:
    #     print(i.id,i.name,i.email,i.message)
    data = {'Signup':signups}
```

```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

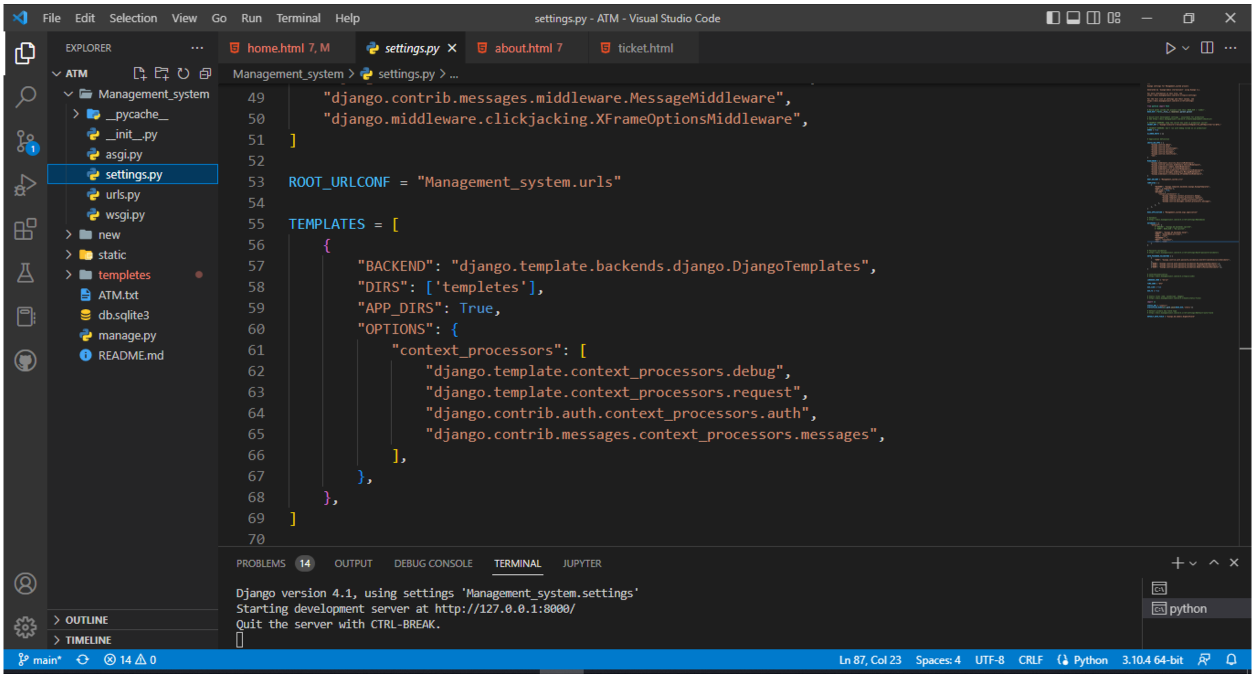
Admin.py



```python
from django.contrib import admin
from .models import Contact, Login, Signup, Ticket
# Register your models here.
admin.site.register(Contact)
admin.site.register(Login)
admin.site.register(Signup)
admin.site.register(Ticket)

```

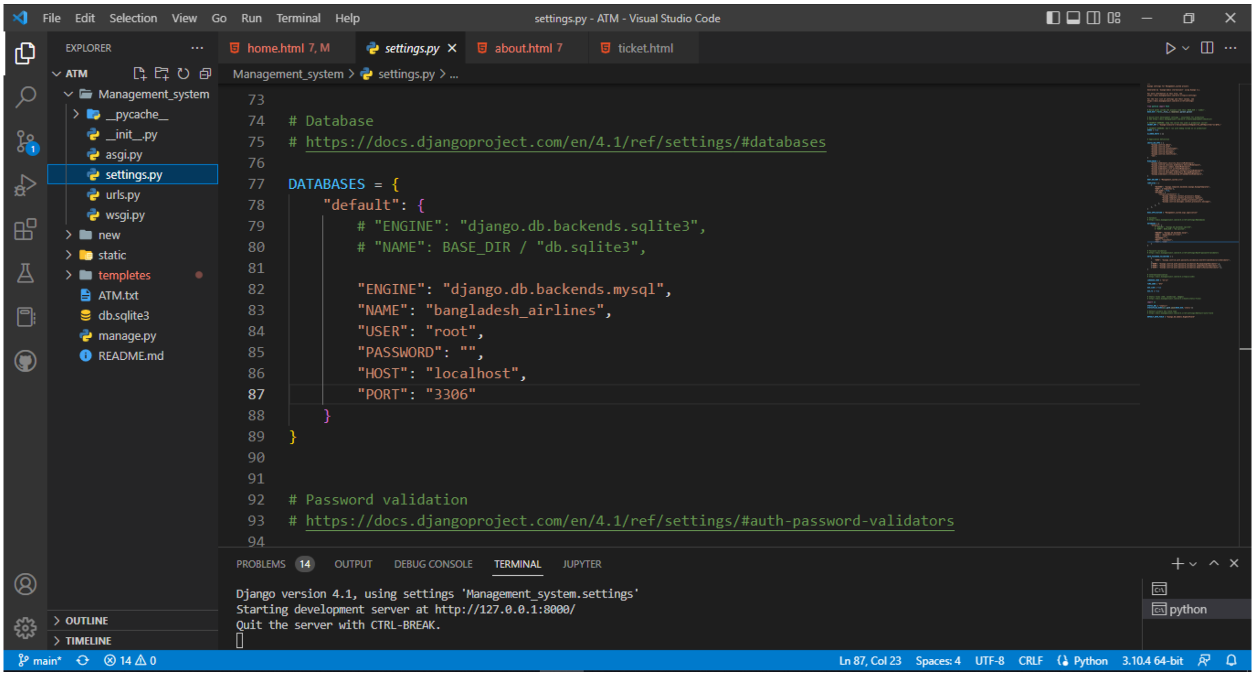
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Settings.py



```
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "Management_system.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": ['templetes'],
        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.debug",
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]

```

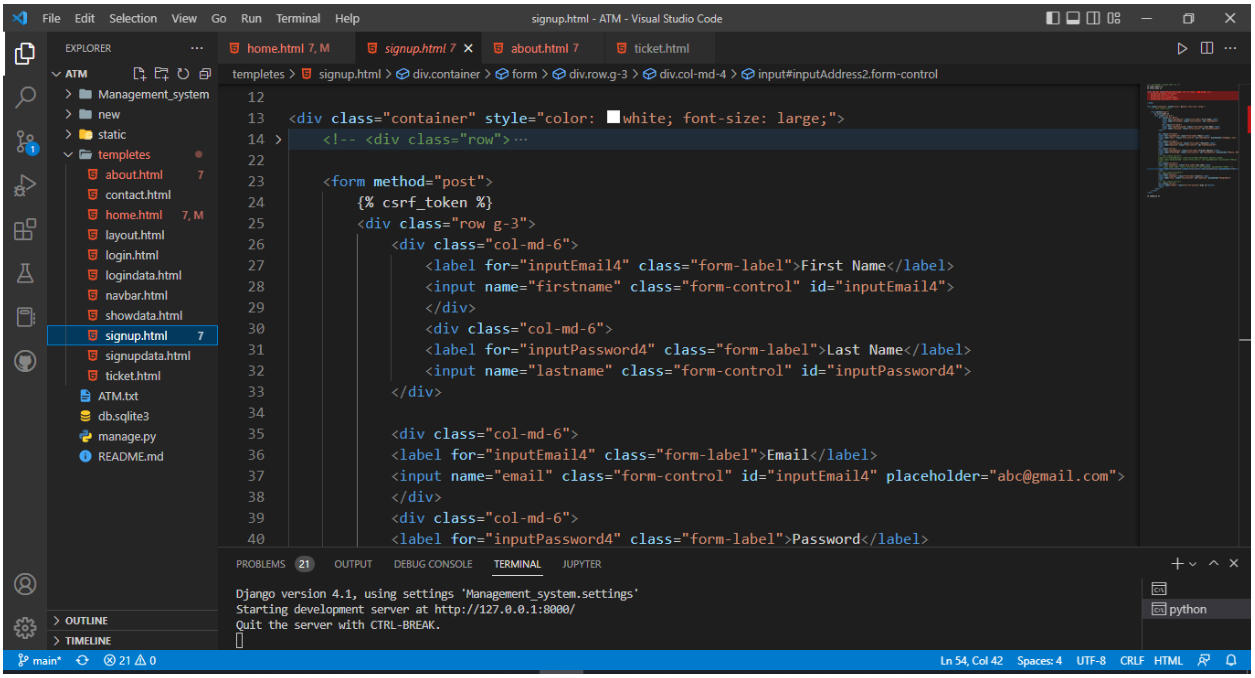
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```

# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    "default": {
        # "ENGINE": "django.db.backends.sqlite3",
        # "NAME": BASE_DIR / "db.sqlite3",

        "ENGINE": "django.db.backends.mysql",
        "NAME": "bangladesh_airlines",
        "USER": "root",
        "PASSWORD": "",
        "HOST": "localhost",
        "PORT": "3306"
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators

```

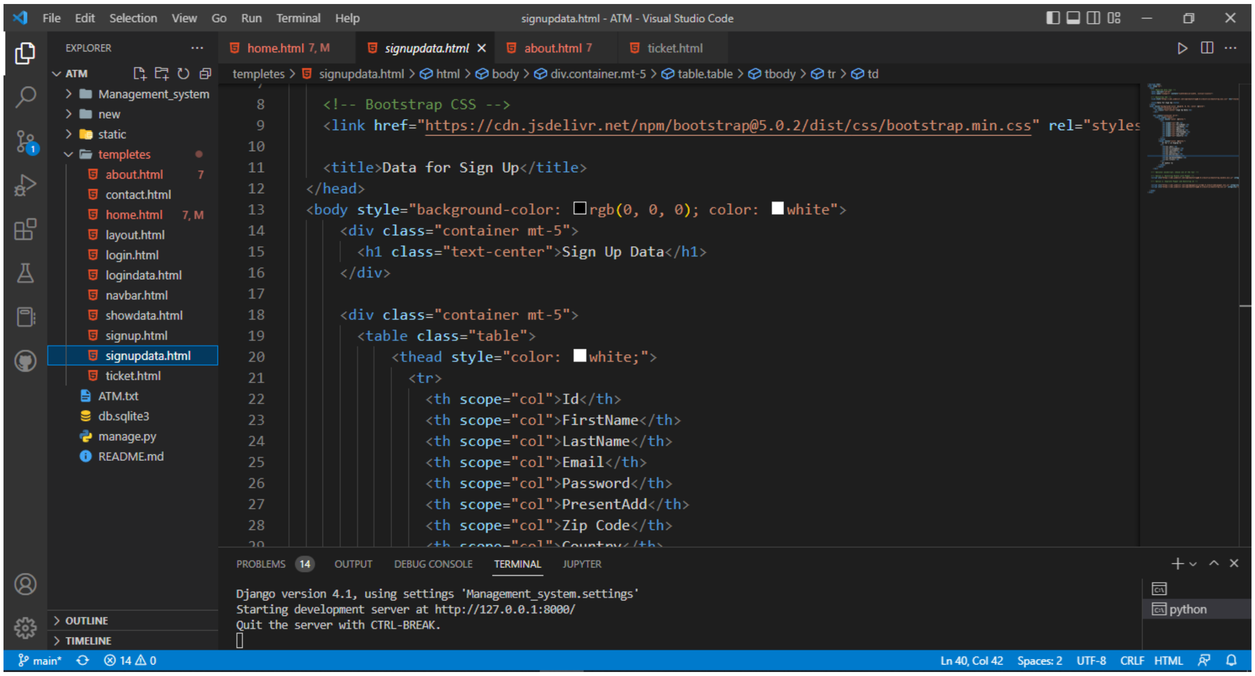
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Templates



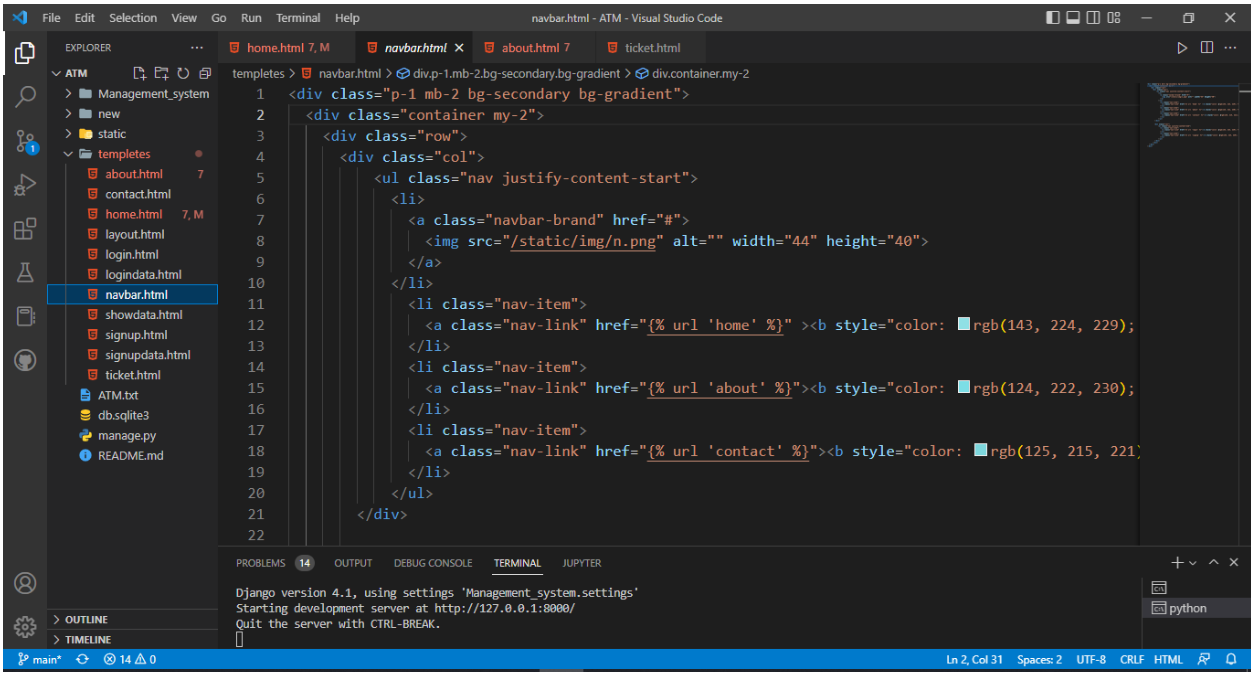
```
<div class="container" style="color: □white; font-size: large;">
    <!-- <div class="row">···

    <form method="post">
        {% csrf_token %}
        <div class="row g-3">
            <div class="col-md-6">
                <label for="inputEmail4" class="form-label">First Name</label>
                <input name="firstname" class="form-control" id="inputEmail4">
                </div>
                <div class="col-md-6">
                <label for="inputPassword4" class="form-label">Last Name</label>
                <input name="lastname" class="form-control" id="inputPassword4">
            </div>

            <div class="col-md-6">
            <label for="inputEmail4" class="form-label">Email</label>
            <input name="email" class="form-control" id="inputEmail4" placeholder="abc@gmail.com">
            </div>
            <div class="col-md-6">
            <label for="inputPassword4" class="form-label">Password</label>
```

```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="styles

  <title>Data for Sign Up</title>
</head>
<body style="background-color: □rgb(0, 0, 0); color: □white">
    <div class="container mt-5">
      <h1 class="text-center">Sign Up Data</h1>
    </div>

    <div class="container mt-5">
      <table class="table">
          <thead style="color: □white;">
            <tr>
              <th scope="col">Id</th>
              <th scope="col">FirstName</th>
              <th scope="col">LastName</th>
              <th scope="col">Email</th>
              <th scope="col">Password</th>
              <th scope="col">PresentAdd</th>
              <th scope="col">Zip Code</th>
              <th scope="col">Country</th>
```

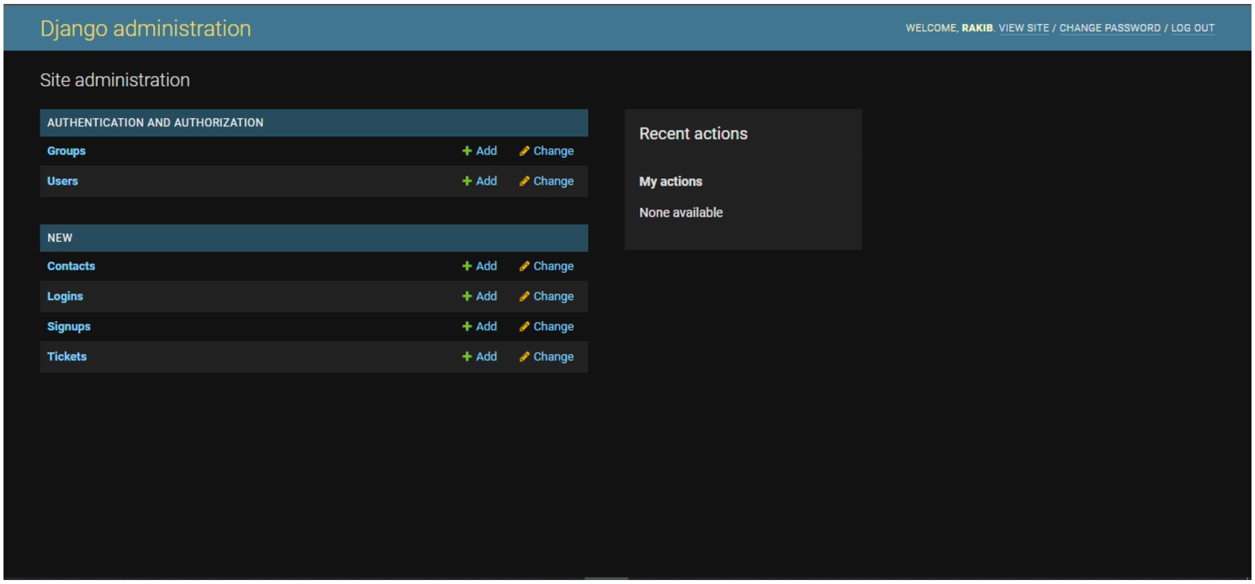
```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```html
<div class="p-1 mb-2 bg-secondary bg-gradient">
  <div class="container my-2">
    <div class="row">
      <div class="col">
          <ul class="nav justify-content-start">
            <li>
              <a class="navbar-brand" href="#">
                <img src="/static/img/n.png" alt="" width="44" height="40">
              </a>
            </li>
              <li class="nav-item">
                <a class="nav-link" href="{% url 'home' %}" ><b style="color: rgb(143, 224, 229);
              </li>
              <li class="nav-item">
                <a class="nav-link" href="{% url 'about' %}"><b style="color: rgb(124, 222, 230);
              </li>
              <li class="nav-item">
                <a class="nav-link" href="{% url 'contact' %}"><b style="color: rgb(125, 215, 221
              </li>
            </ul>
        </div>

```

```
Django version 4.1, using settings 'Management_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Admin site



Site administration

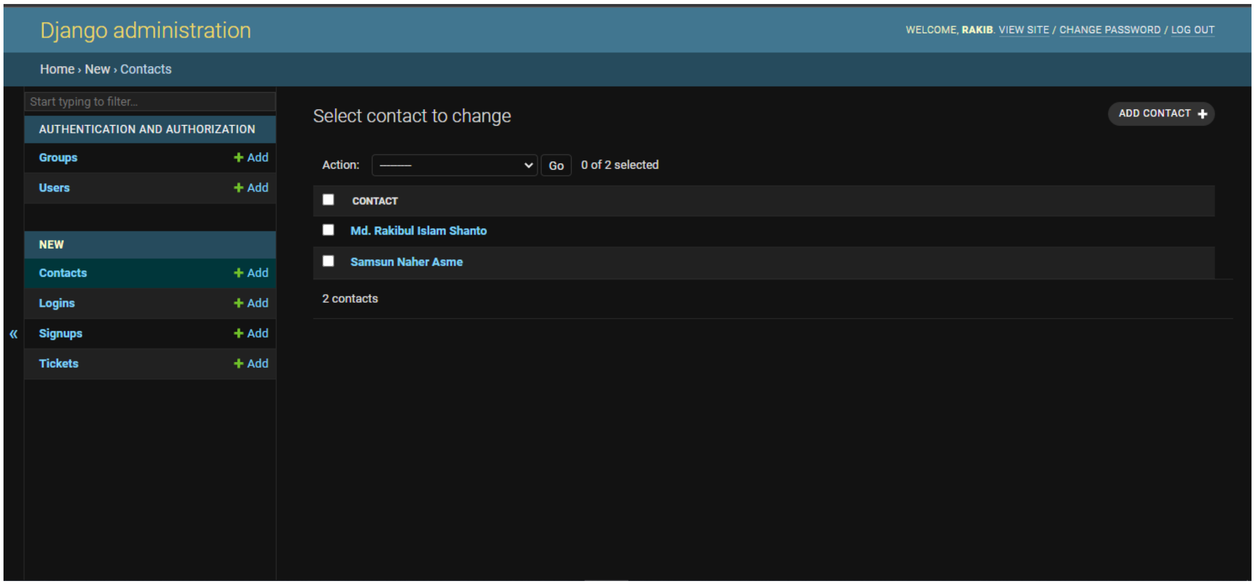
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

NEW		
Contacts	+ Add	Change
Logins	+ Add	Change
Signups	+ Add	Change
Tickets	+ Add	Change

Recent actions

My actions

None available

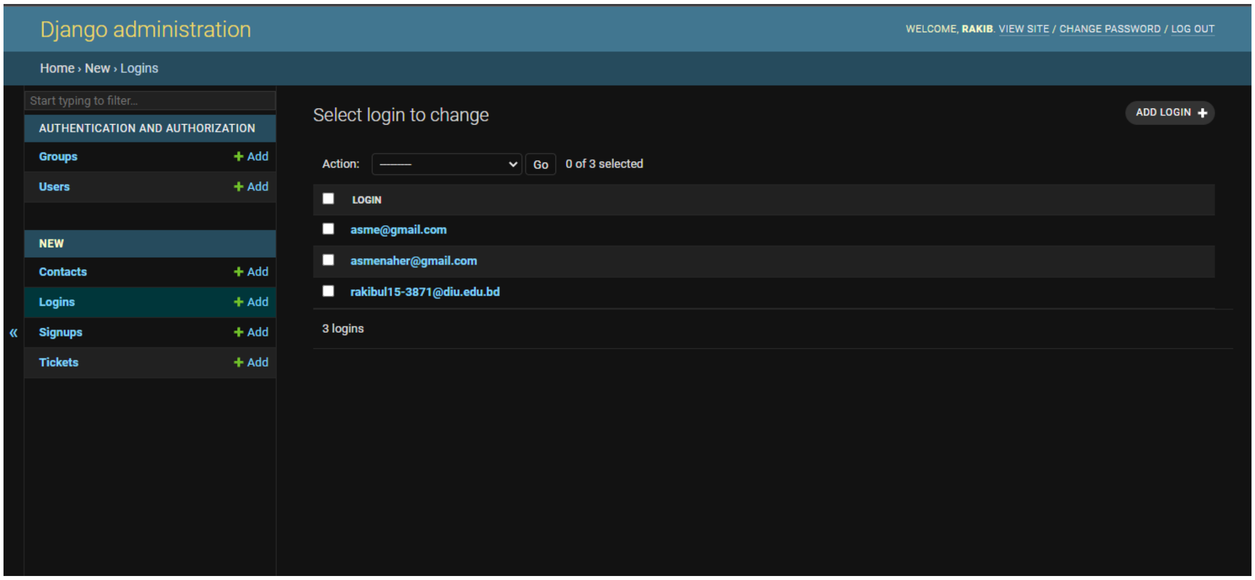


Select contact to change

0 of 2 selected

□	CONTACT
□	Md. Rakibul Islam Shanto
□	Samsun Naher Asme

2 contacts



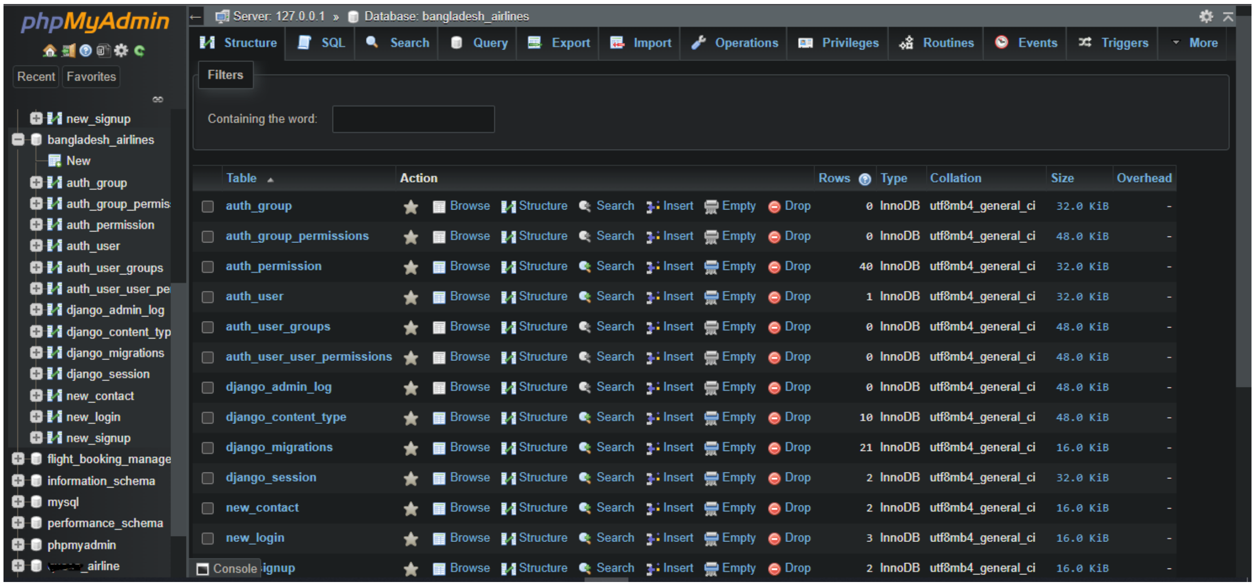
Select login to change

0 of 3 selected

□	LOGIN
□	asme@gmail.com
□	asmenaher@gmail.com
□	rakibul15-3871@diu.edu.bd

3 logins

MySQL



phpMyAdmin

Recent Favorites

- new_signup
- bangladesh_airlines
  - New
  - auth_group
  - auth_group_permis
  - auth_permission
  - auth_user
  - auth_user_groups
  - auth_user_user_pe
  - django_admin_log
  - django_content_typ
  - django_migrations
  - django_session
  - new_contact
  - new_login
  - new_signup
- flight_booking_manage
- information_schema
- mysql
- performance_schema
- phpmyadmin
- [redacted] airline

Server: 127.0.0.1 » Database: bangladesh_airlines

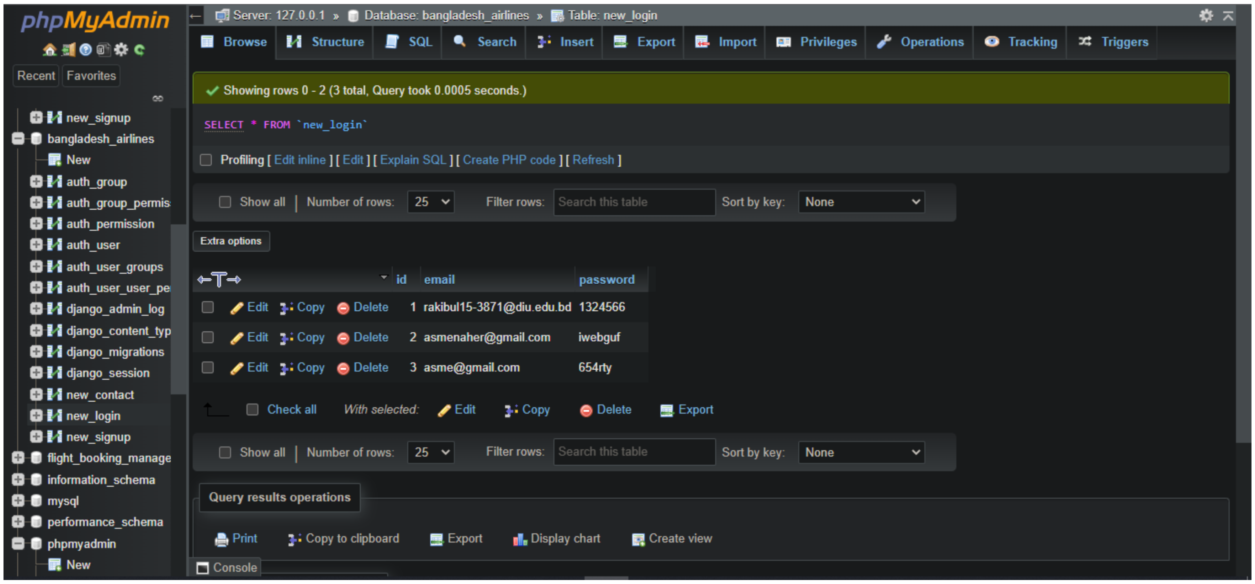
Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events | Triggers | More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
auth_group	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_group_permissions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
auth_permission	★ Browse Structure Search Insert Empty Drop	40	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_user	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_user_groups	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
auth_user_user_permissions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_admin_log	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_content_type	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_migrations	★ Browse Structure Search Insert Empty Drop	21	InnoDB	utf8mb4_general_ci	16.0 KiB	-
django_session	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
new_contact	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
new_login	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
ignup	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-

Console



phpMyAdmin

Recent Favorites

- new_signup
- bangladesh_airlines
  - New
  - auth_group
  - auth_group_permis
  - auth_permission
  - auth_user
  - auth_user_groups
  - auth_user_user_pe
  - django_admin_log
  - django_content_typ
  - django_migrations
  - django_session
  - new_contact
  - new_login
  - new_signup
- flight_booking_manage
- information_schema
- mysql
- performance_schema
- phpmyadmin
  - New

Server: 127.0.0.1 » Database: bangladesh_airlines » Table: new_login

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

✓ Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT * FROM `new_login`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	id	email	password
□ Edit Copy Delete	1	rakibul15-3871@diu.edu.bd	1324566
□ Edit Copy Delete	2	asmenaher@gmail.com	iwebguf
□ Edit Copy Delete	3	asme@gmail.com	654rty

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

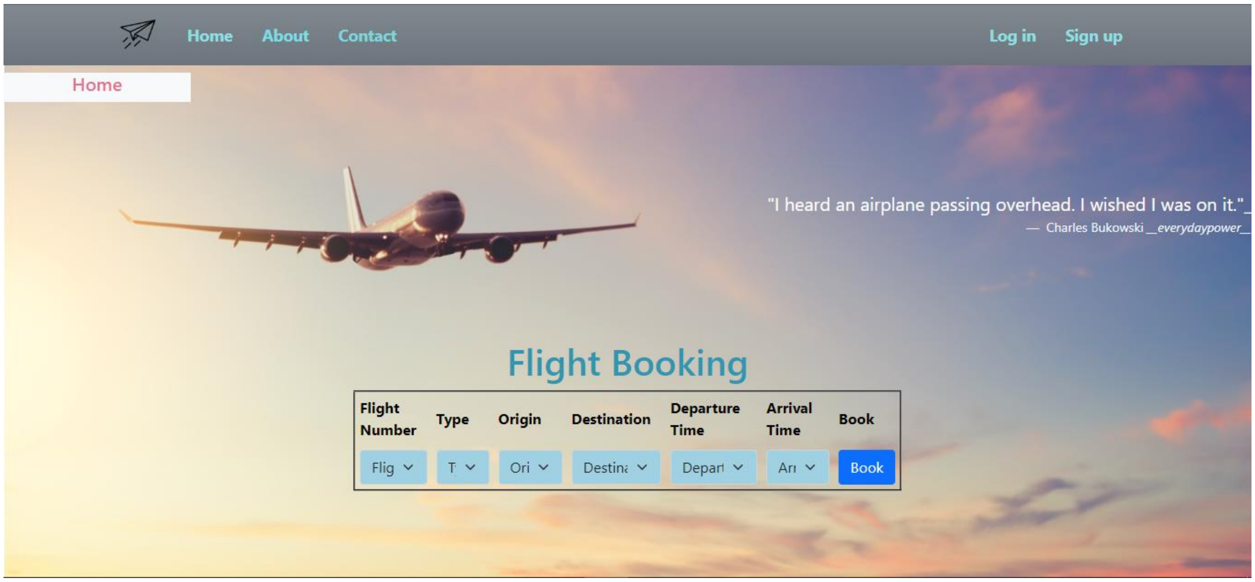
Print Copy to clipboard Export Display chart Create view

Console

Frontend

Django templates, which the framework converts by request to HTML pages that our users will be able to interact with, are used to develop the front end of our web application.

Home



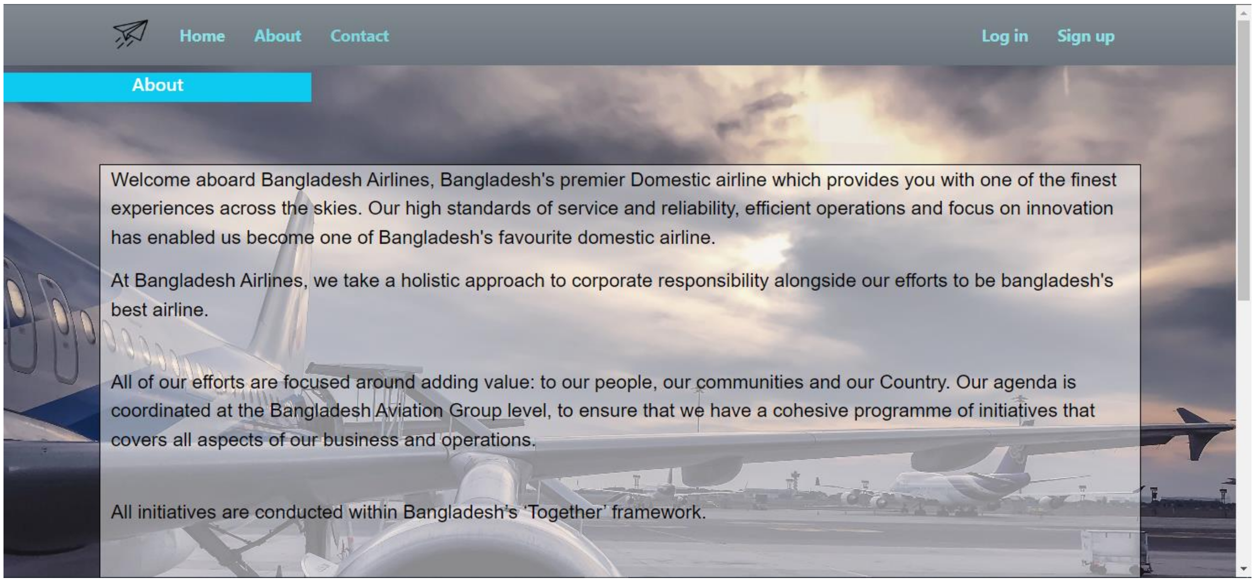
Home

"I heard an airplane passing overhead. I wished I was on it."_

— Charles Bukowski _*everydaypower*_

Flight Booking

Flight Number	Type	Origin	Destination	Departure Time	Arrival Time	Book
Flig	T	Ori	Destin	Depart	Arr	Book



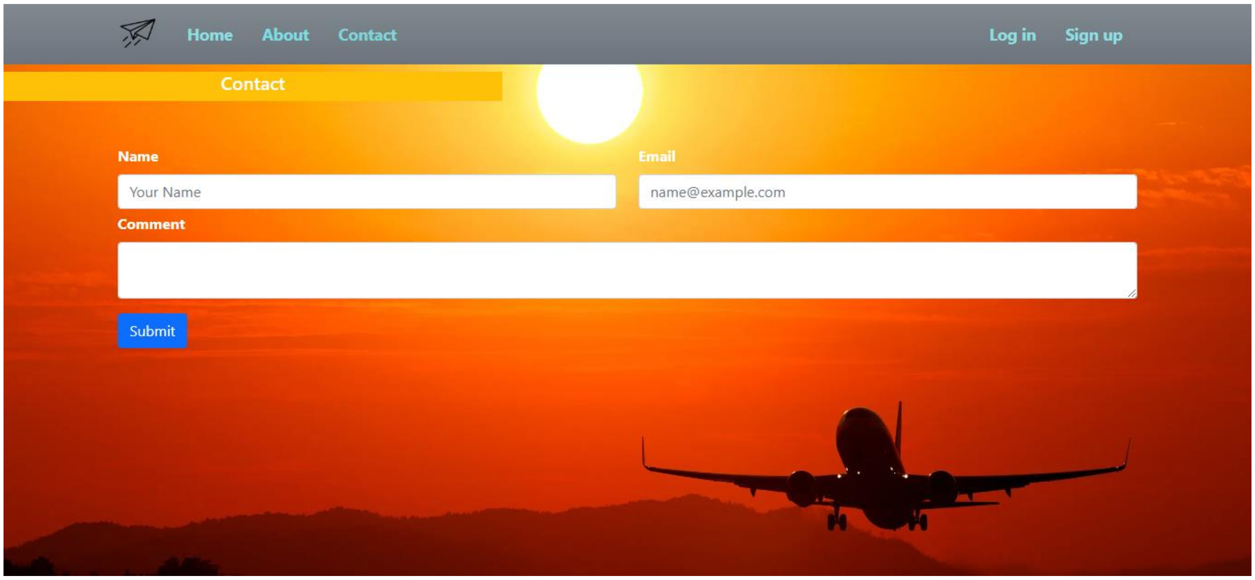
About

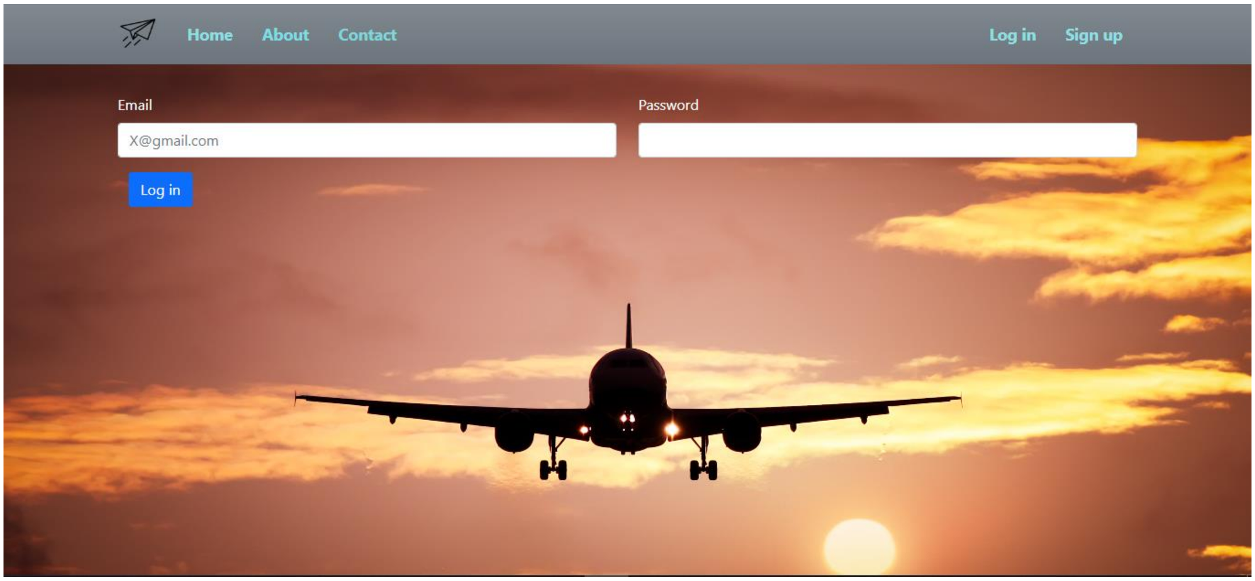
Welcome aboard Bangladesh Airlines, Bangladesh's premier Domestic airline which provides you with one of the finest experiences across the skies. Our high standards of service and reliability, efficient operations and focus on innovation has enabled us become one of Bangladesh's favourite domestic airline.

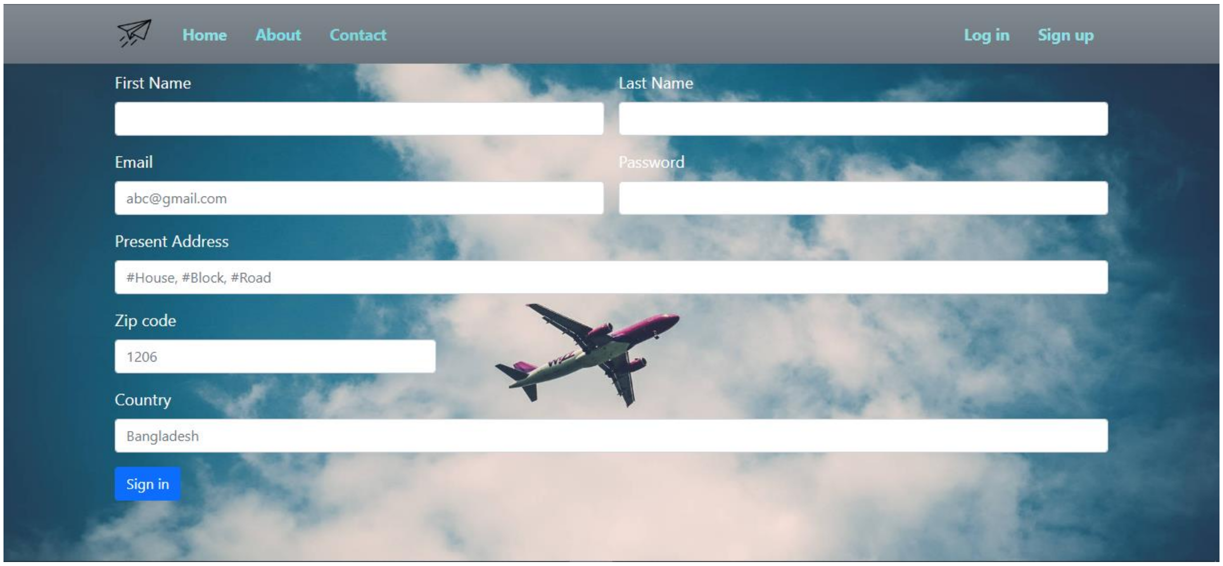
At Bangladesh Airlines, we take a holistic approach to corporate responsibility alongside our efforts to be bangladesh's best airline.

All of our efforts are focused around adding value: to our people, our communities and our Country. Our agenda is coordinated at the Bangladesh Aviation Group level, to ensure that we have a cohesive programme of initiatives that covers all aspects of our business and operations.

All initiatives are conducted within Bangladesh's 'Together' framework.


Home
About
Contact
Log in
Sign up
Contact
Name
Your Name
Email
name@example.com
Comment
Submit


Home
About
Contact
Log in
Sign up
Email
X@gmail.com
Password
Log in



First Name

Last Name

Email

abc@gmail.com

Password

Present Address

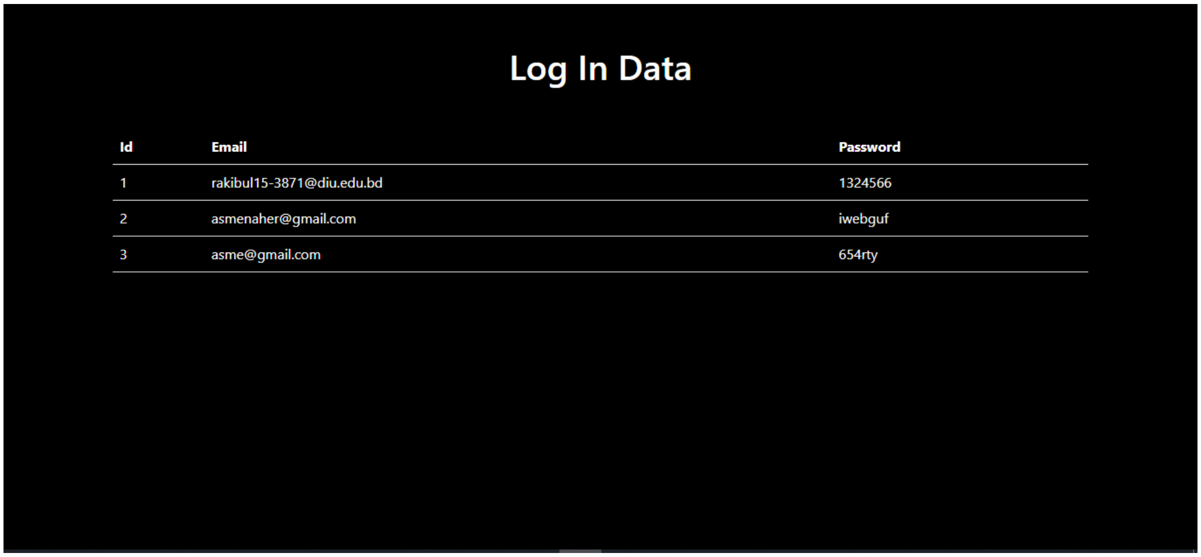
#House, #Block, #Road

Zip code

1206

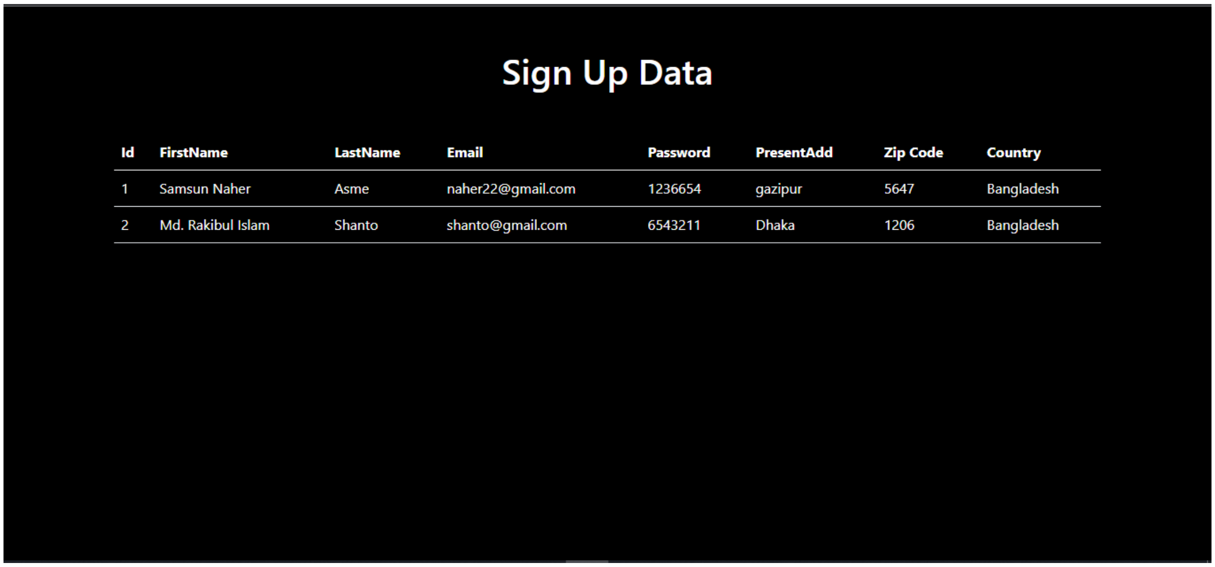
Country

Bangladesh



Log In Data

Id	Email	Password
1	rakibul15-3871@diu.edu.bd	1324566
2	asmenaher@gmail.com	iwebguf
3	asme@gmail.com	654rty



Sign Up Data

Id	FirstName	LastName	Email	Password	PresentAdd	Zip Code	Country
1	Samsun Naher	Asme	naher22@gmail.com	1236654	gazipur	5647	Bangladesh
2	Md. Rakibul Islam	Shanto	shanto@gmail.com	6543211	Dhaka	1206	Bangladesh

Conclusion

The possibility by our esteemed professors to complete this assignment is a crucial component of our course "OOP 2." We were able to create something on our own thanks to this project. To develop the project, we employed a few software programs and design ideas. We were greatly motivated to construct further projects of this nature by the process of building this one. Many fans of web programming now have a new job choice thanks to Django.

Last but not least, we are grateful for this project since it allowed us to learn a lot of things that will be useful to us in the future.

Github Link

https://github.com/cryptic91/Bangladesh_Airlines

------------------------------ 0 ------------------------------