



## 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

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

## **Project Name**

### **Bangladesh Airlines**

## **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

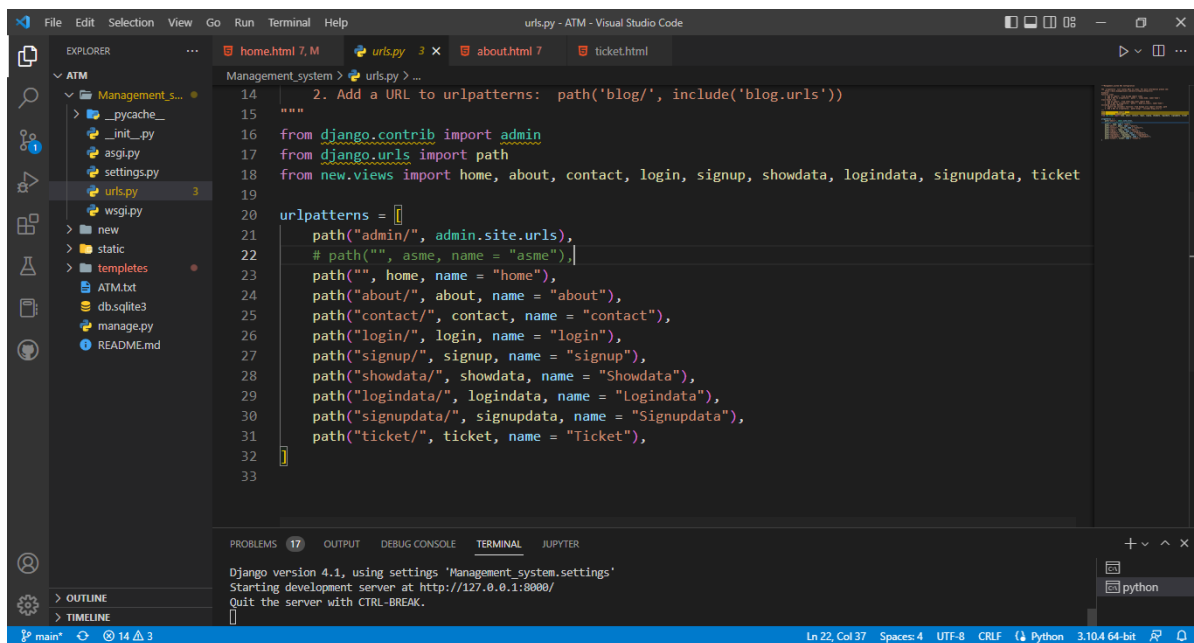
This project was created utilizing software like VSCode, Annaconda, Utilize a database, Create a table, add an image, use HTML and CSS, utilizing Python, HTML, Django, and Bootstrap.

## 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



```
File Edit Selection View Go Run Terminal Help
urls.py - ATM - Visual Studio Code

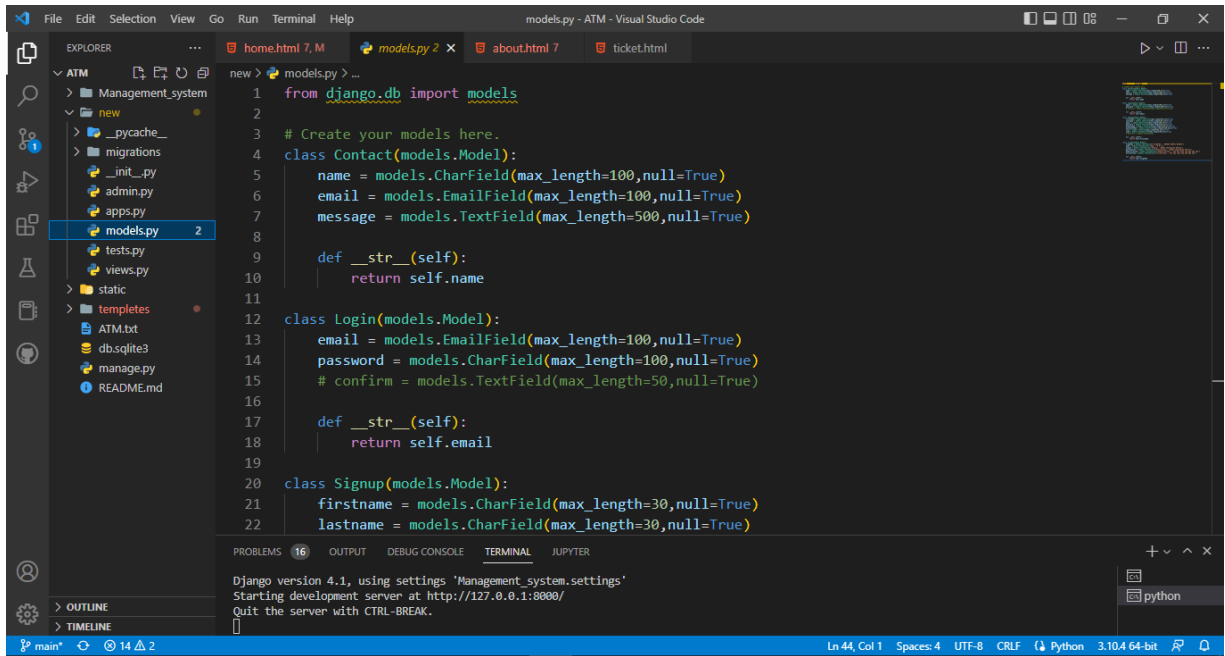
EXPLORER
  ATM
    Management_s...
    __pycache__
    _init_.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
    new
    static
    templates
    ATM.txt
    db.sqlite3
    manage.py
    README.md

14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15
16 from django.contrib import admin
17 from django.urls import path
18 from new.views import home, about, contact, login, signup, showdata, logindata, signupdata, ticket
19
20 urlpatterns = []
21 path("admin/", admin.site.urls),
22 # path("", asme, name = "asme"),
23 path("", home, name = "home"),
24 path("about/", about, name = "about"),
25 path("contact/", contact, name = "contact"),
26 path("login/", login, name = "login"),
27 path("signup/", signup, name = "signup"),
28 path("showdata/", showdata, name = "Showdata"),
29 path("logindata/", logindata, name = "Logindata"),
30 path("signupdata/", signupdata, name = "Signupdata"),
31 path("ticket/", ticket, name = "Ticket"),
32
33

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
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.

main* 14 3 Ln 22, Col 37 Spaces: 4 UTF-8 CRLF Python 3.10.4 64-bit
```

# Models.py



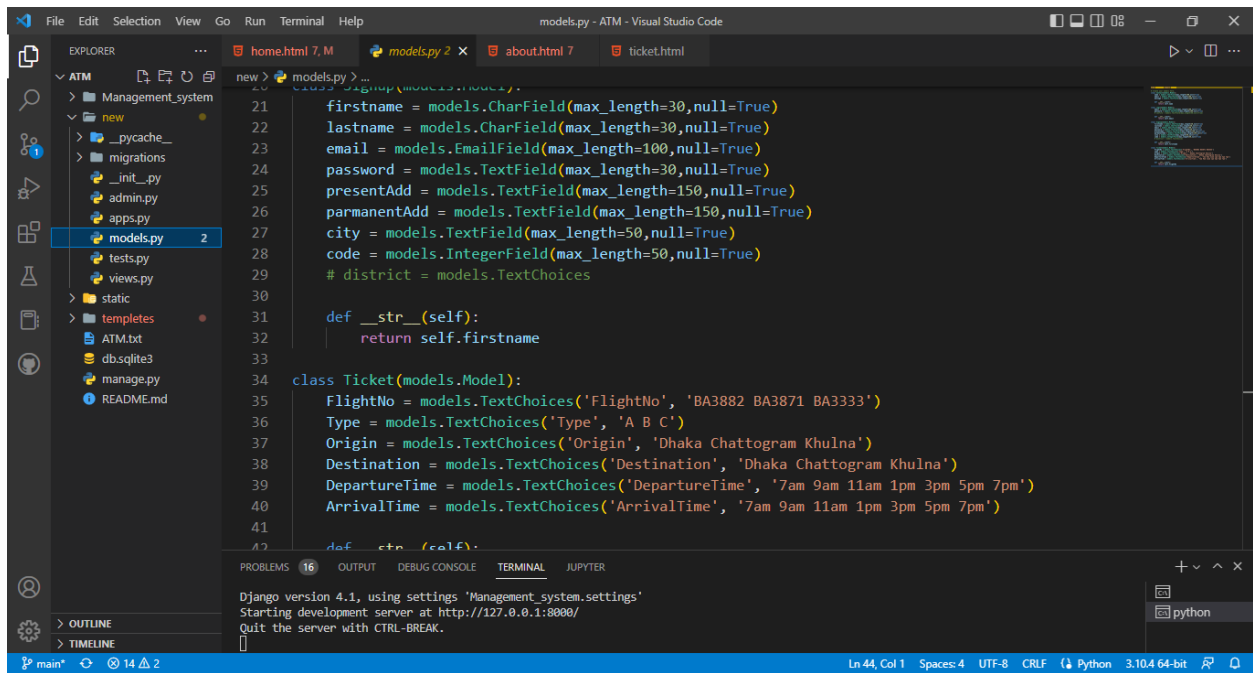
The screenshot shows the Visual Studio Code editor with the 'models.py' file open. The file contains the following code:

```
1 from django.db import models
2
3 # Create your models here.
4 class Contact(models.Model):
5     name = models.CharField(max_length=100,null=True)
6     email = models.EmailField(max_length=100,null=True)
7     message = models.TextField(max_length=500,null=True)
8
9     def __str__(self):
10         return self.name
11
12 class Login(models.Model):
13     email = models.EmailField(max_length=100,null=True)
14     password = models.CharField(max_length=100,null=True)
15     # confirm = models.TextField(max_length=50,null=True)
16
17     def __str__(self):
18         return self.email
19
20 class Signup(models.Model):
21     firstname = models.CharField(max_length=30,null=True)
22     lastname = models.CharField(max_length=30,null=True)
```

The Explorer sidebar on the left shows the project structure for 'ATM', including files like 'Management\_system', 'migrations', 'admin.py', 'apps.py', 'models.py', 'tests.py', 'views.py', 'static', 'templates', 'ATM.txt', 'db.sqlite3', 'manage.py', and 'README.md'. The 'models.py' file is selected and highlighted.

The bottom panel shows the 'TERMINAL' output with the following text:

```
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.
```



The screenshot shows the Visual Studio Code editor with the 'models.py' file open, displaying the updated code. The 'Signup' class has been expanded with more fields, and a new 'Ticket' class has been added.

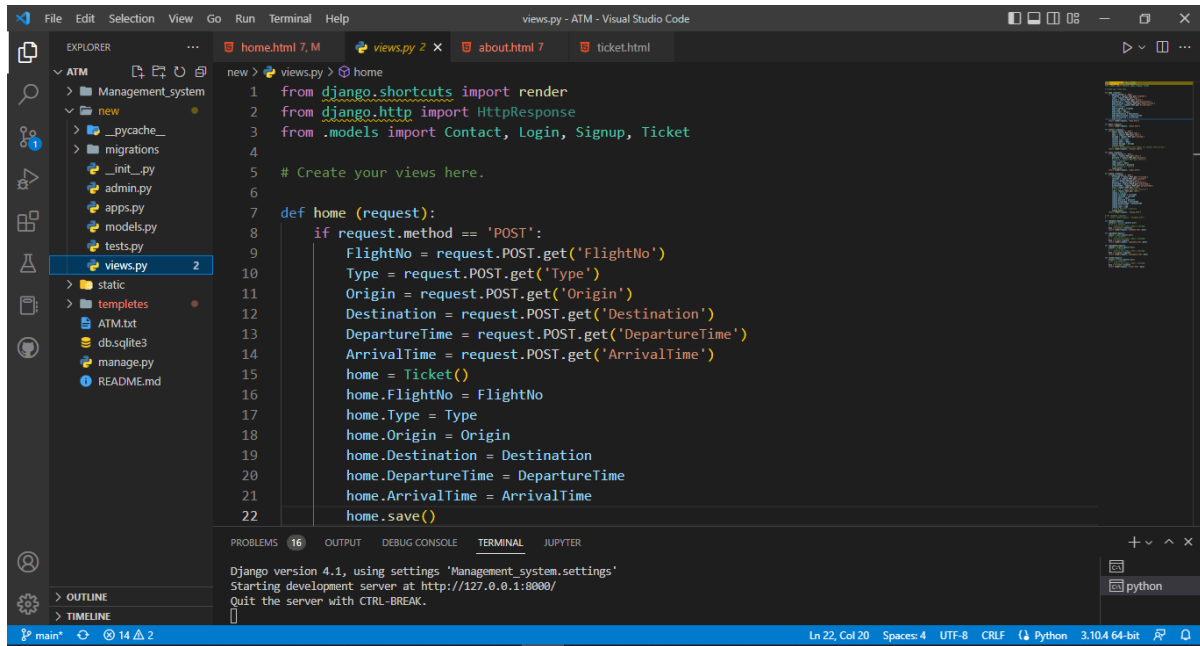
```
20 class Signup(models.Model):
21     firstname = models.CharField(max_length=30,null=True)
22     lastname = models.CharField(max_length=30,null=True)
23     email = models.EmailField(max_length=100,null=True)
24     password = models.TextField(max_length=30,null=True)
25     presentAdd = models.TextField(max_length=150,null=True)
26     parmanentAdd = models.TextField(max_length=150,null=True)
27     city = models.TextField(max_length=50,null=True)
28     code = models.IntegerField(max_length=50,null=True)
29     # district = models.TextChoices
30
31     def __str__(self):
32         return self.firstname
33
34 class Ticket(models.Model):
35     FlightNo = models.TextChoices('FlightNo', 'BA3882 BA3871 BA3333')
36     Type = models.TextChoices('Type', 'A B C')
37     Origin = models.TextChoices('Origin', 'Dhaka Chattogram Khulna')
38     Destination = models.TextChoices('Destination', 'Dhaka Chattogram Khulna')
39     DepartureTime = models.TextChoices('DepartureTime', '7am 9am 11am 1pm 3pm 5pm 7pm')
40     Arrivalltime = models.TextChoices('Arrivalltime', '7am 9am 11am 1pm 3pm 5pm 7pm')
41
42     def __str__(self):
```

The Explorer sidebar on the left shows the project structure for 'ATM', including files like 'Management\_system', 'migrations', 'admin.py', 'apps.py', 'models.py', 'tests.py', 'views.py', 'static', 'templates', 'ATM.txt', 'db.sqlite3', 'manage.py', and 'README.md'. The 'models.py' file is selected and highlighted.

The bottom panel shows the 'TERMINAL' output with the following text:

```
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.
```

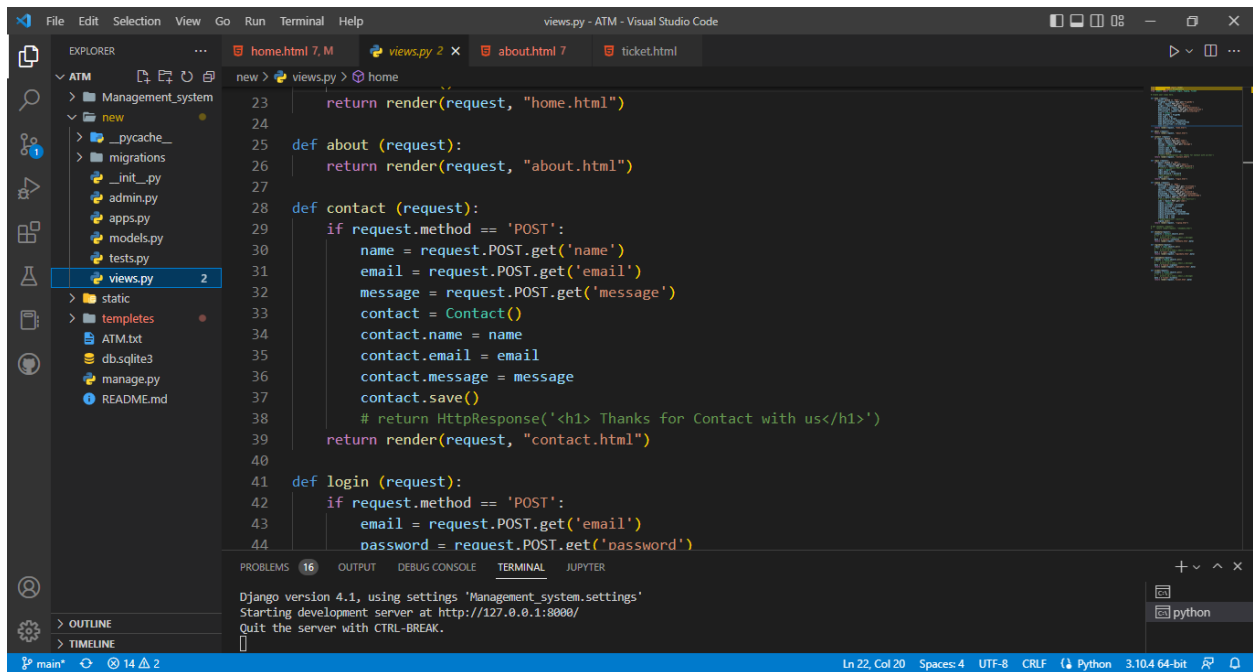
# Views.py



The screenshot shows the Visual Studio Code interface with the 'ATM' project open. The Explorer sidebar on the left shows the project structure, including 'Management\_system', 'new', and 'static' folders. The 'views.py' file is selected and open in the editor. The code in 'views.py' is as follows:

```
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 from .models import Contact, Login, Signup, Ticket
4
5 # Create your views here.
6
7 def home (request):
8     if request.method == 'POST':
9         FlightNo = request.POST.get('FlightNo')
10        Type = request.POST.get('Type')
11        Origin = request.POST.get('Origin')
12        Destination = request.POST.get('Destination')
13        DepartureTime = request.POST.get('DepartureTime')
14        ArrivalTime = request.POST.get('ArrivalTime')
15        home = Ticket()
16        home.FlightNo = FlightNo
17        home.Type = Type
18        home.Origin = Origin
19        home.Destination = Destination
20        home.DepartureTime = DepartureTime
21        home.ArrivalTime = ArrivalTime
22        home.save()
```

The terminal at the bottom shows the Django version 4.1 and the starting of the development server at http://127.0.0.1:8000/.



The screenshot shows the Visual Studio Code interface with the 'ATM' project open. The 'views.py' file is selected and open in the editor. The code in 'views.py' is as follows:

```
23     return render(request, "home.html")
24
25 def about (request):
26     return render(request, "about.html")
27
28 def contact (request):
29     if request.method == 'POST':
30         name = request.POST.get('name')
31         email = request.POST.get('email')
32         message = request.POST.get('message')
33         contact = Contact()
34         contact.name = name
35         contact.email = email
36         contact.message = message
37         contact.save()
38         # return HttpResponseRedirect('<h1> Thanks for Contact with us</h1>')
39     return render(request, "contact.html")
40
41 def login (request):
42     if request.method == 'POST':
43         email = request.POST.get('email')
44         password = request.POST.get('password')
```

The terminal at the bottom shows the Django version 4.1 and the starting of the development server at http://127.0.0.1:8000/.

```
77 # def showdata (request):
78 #     return render(request, "showdata.html")
79
80 def showdata(request):
81     contacts = Contact.objects.all()
82     # for i in contacts:
83     #     print(i.id,i.name,i.email,i.message)
84     data = {'Contact':contacts}
85     return render(request,'showdata.html',data)
86
87 def logindata(request):
88     logins = Login.objects.all()
89     # for i in contacts:
90     #     print(i.id,i.name,i.email,i.message)
91     data = {'Login':logins}
92     return render(request,'logindata.html',data)
93
94 def signupdata(request):
95     signups = Signup.objects.all()
96     # for i in contacts:
97     #     print(i.id,i.name,i.email,i.message)
98     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.

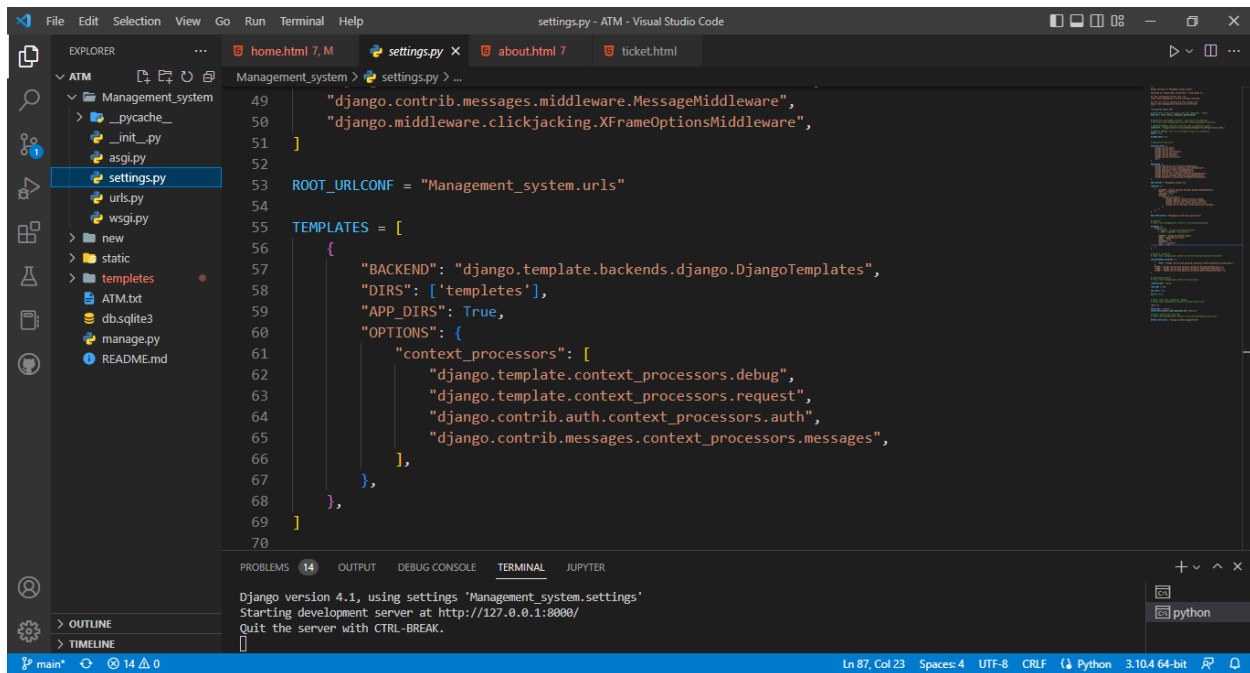
## Admin.py

```
1 from django.contrib import admin
2 from .models import Contact, Login, Signup, Ticket
3 # Register your models here.
4 admin.site.register(Contact)
5 admin.site.register(Login)
6 admin.site.register(Signup)
7 admin.site.register(Ticket)
8
```

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.

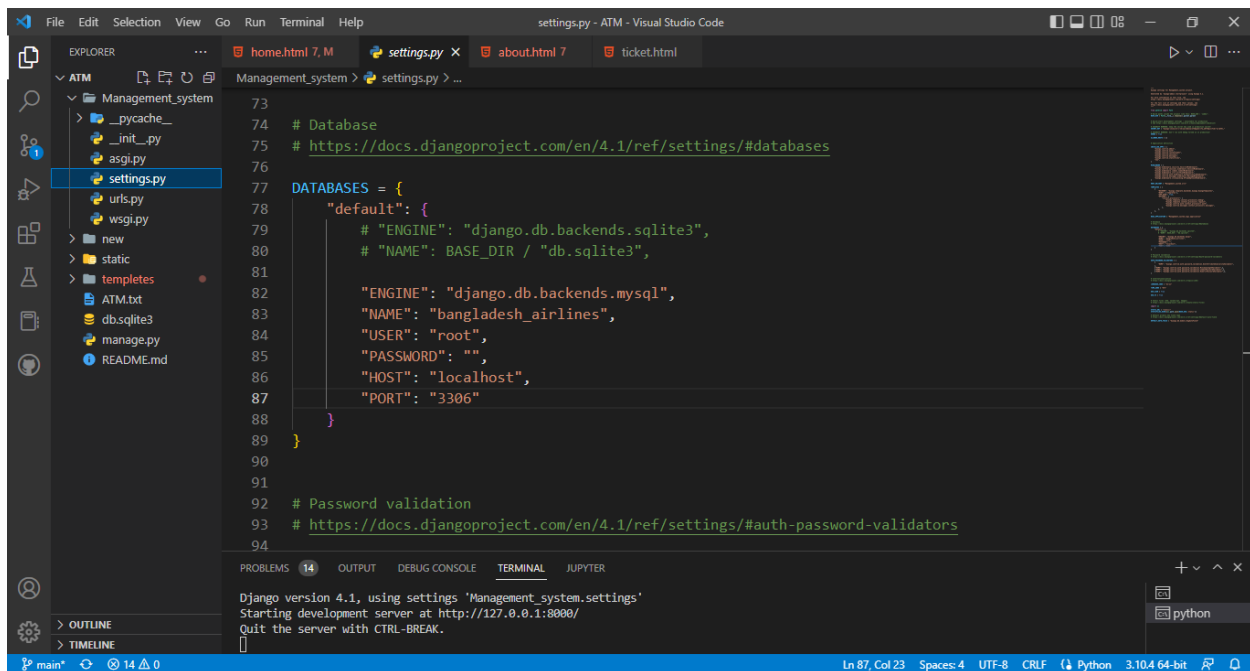


# Settings.py



```
49 "django.contrib.messages.middleware.MessageMiddleware",
50 "django.middleware.clickjacking.XFrameOptionsMiddleware",
51 ]
52
53 ROOT_URLCONF = "Management_system.urls"
54
55 TEMPLATES = [
56     {
57         "BACKEND": "django.template.backends.django.DjangoTemplates",
58         "DIRS": ['templates'],
59         "APP_DIRS": True,
60         "OPTIONS": {
61             "context_processors": [
62                 "django.template.context_processors.debug",
63                 "django.template.context_processors.request",
64                 "django.contrib.auth.context_processors.auth",
65                 "django.contrib.messages.context_processors.messages",
66             ],
67         },
68     },
69 ]
70
```

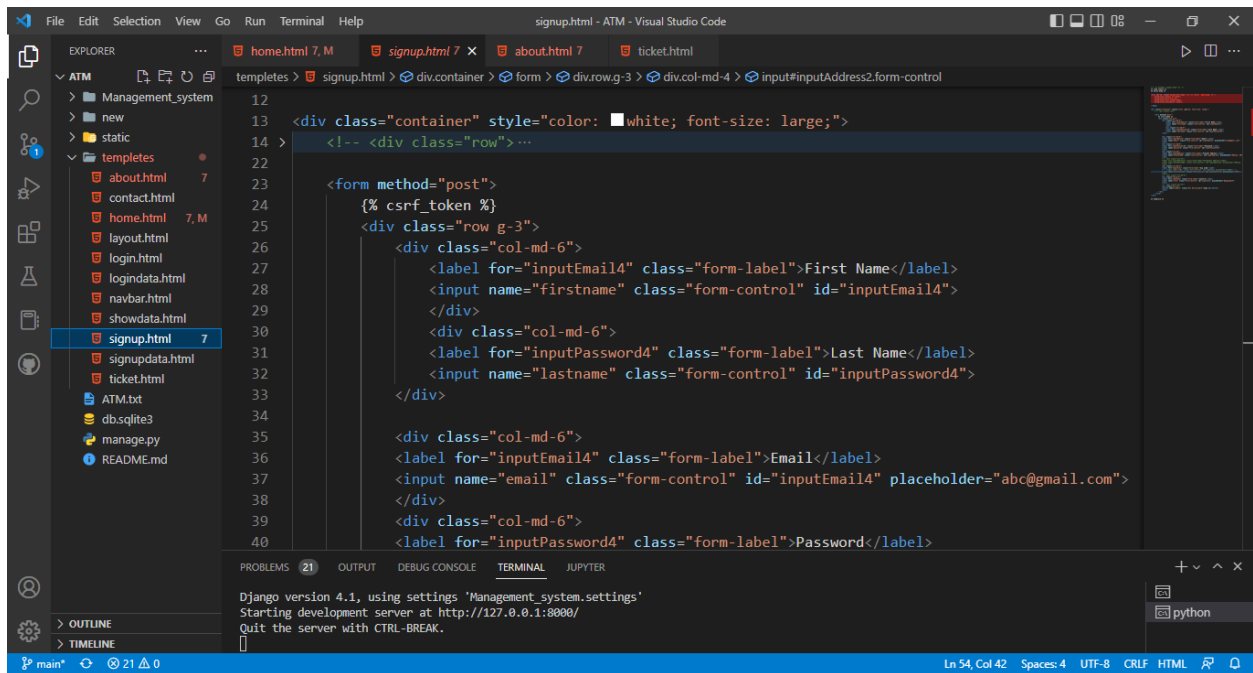
Django version 4.1, using settings 'Management\_system.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.



```
73
74 # Database
75 # https://docs.djangoproject.com/en/4.1/ref/settings/#databases
76
77 DATABASES = {
78     "default": {
79         # "ENGINE": "django.db.backends.sqlite3",
80         # "NAME": BASE_DIR / "db.sqlite3",
81
82         "ENGINE": "django.db.backends.mysql",
83         "NAME": "bangladesh_airlines",
84         "USER": "root",
85         "PASSWORD": "",
86         "HOST": "localhost",
87         "PORT": "3306"
88     }
89 }
90
91
92 # Password validation
93 # https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators
94
```

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.

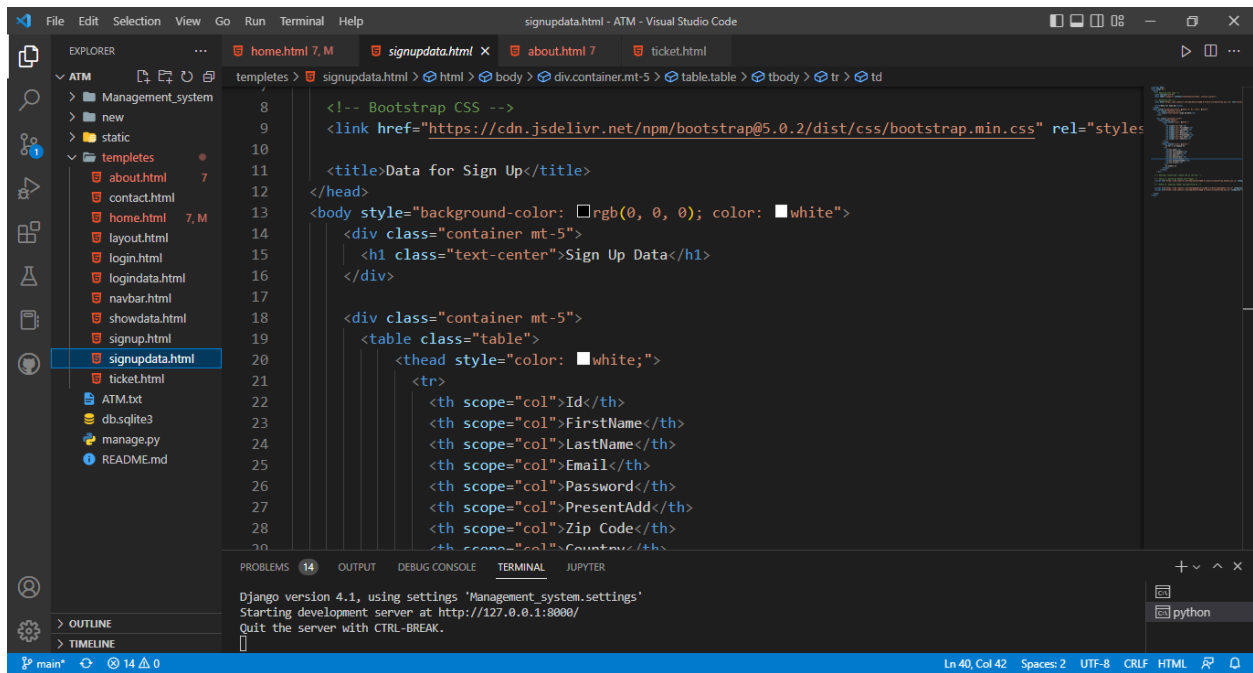
# Templates



```
12
13 <div class="container" style="color: white; font-size: large;">
14 <!-- <div class="row"> ...
22
23 <form method="post">
24     {% csrf_token %}
25     <div class="row g-3">
26         <div class="col-md-6">
27             <label for="inputEmail4" class="form-label">First Name</label>
28             <input name="firstname" class="form-control" id="inputEmail4">
29         </div>
30         <div class="col-md-6">
31             <label for="inputPassword4" class="form-label">Last Name</label>
32             <input name="lastname" class="form-control" id="inputPassword4">
33         </div>
34     </div>
35     <div class="col-md-6">
36         <label for="inputEmail4" class="form-label">Email</label>
37         <input name="email" class="form-control" id="inputEmail4" placeholder="abc@gmail.com">
38     </div>
39     <div class="col-md-6">
40         <label for="inputPassword4" class="form-label">Password</label>
```

PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

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.



```
8 <!-- Bootstrap CSS -->
9 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
10
11 <title>Data for Sign Up</title>
12 </head>
13 <body style="background-color: rgb(0, 0, 0); color: white">
14     <div class="container mt-5">
15         <h1 class="text-center">Sign Up Data</h1>
16     </div>
17
18     <div class="container mt-5">
19         <table class="table">
20             <thead style="color: white;">
21                 <tr>
22                     <th scope="col">Id</th>
23                     <th scope="col">FirstName</th>
24                     <th scope="col">LastName</th>
25                     <th scope="col">Email</th>
26                     <th scope="col">Password</th>
27                     <th scope="col">PresentAdd</th>
28                     <th scope="col">Zip Code</th>
29                     <th scope="col">Country</th>
30                 </tr>
31             </thead>
32             <tbody>
33                 <tr>
34                     <td>1</td>
35                     <td>John</td>
36                     <td>Doe</td>
37                     <td>john.doe@gmail.com</td>
38                     <td>123456</td>
39                     <td>123 Main St</td>
40                     <td>12345</td>
41                     <td>USA</td>
42                 </tr>
43             </tbody>
44         </table>
45     </div>
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

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.

```
1 <div class="p-1 mb-2 bg-secondary bg-gradient">
2   <div class="container my-2">
3     <div class="row">
4       <div class="col">
5         <ul class="nav justify-content-start">
6           <li>
7             <a class="navbar-brand" href="#">
8               
9             </a>
10          </li>
11          <li class="nav-item">
12            <a class="nav-link" href="{% url 'home' %}" ><b style="color: rgb(143, 224, 229);">
13          </li>
14          <li class="nav-item">
15            <a class="nav-link" href="{% url 'about' %}"><b style="color: rgb(124, 222, 230);">
16          </li>
17          <li class="nav-item">
18            <a class="nav-link" href="{% url 'contact' %}"><b style="color: rgb(125, 215, 221);">
19          </li>
20        </ul>
21      </div>
22    </div>
  </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

Django administration

WELCOME, RAKIB / VIEW SITE / CHANGE PASSWORD / LOG OUT

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

Django administration

WELCOME, RAKIB / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home » New » Contacts

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

NEW

Contacts + Add

Logins + Add

Signups + Add

Tickets + Add

Select contact to change

ADD CONTACT +

Action:  Go 0 of 2 selected

☐ CONTACT

☐ Md. Rakibul Islam Shanto

☐ Samsun Naher Asme

2 contacts

Django administration

WELCOME, RAKIB / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home » New » Logins

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

NEW

Contacts + Add

Logins + Add

Signups + Add

Tickets + Add

Select login to change

ADD LOGIN +

Action:  Go 0 of 3 selected

☐ LOGIN

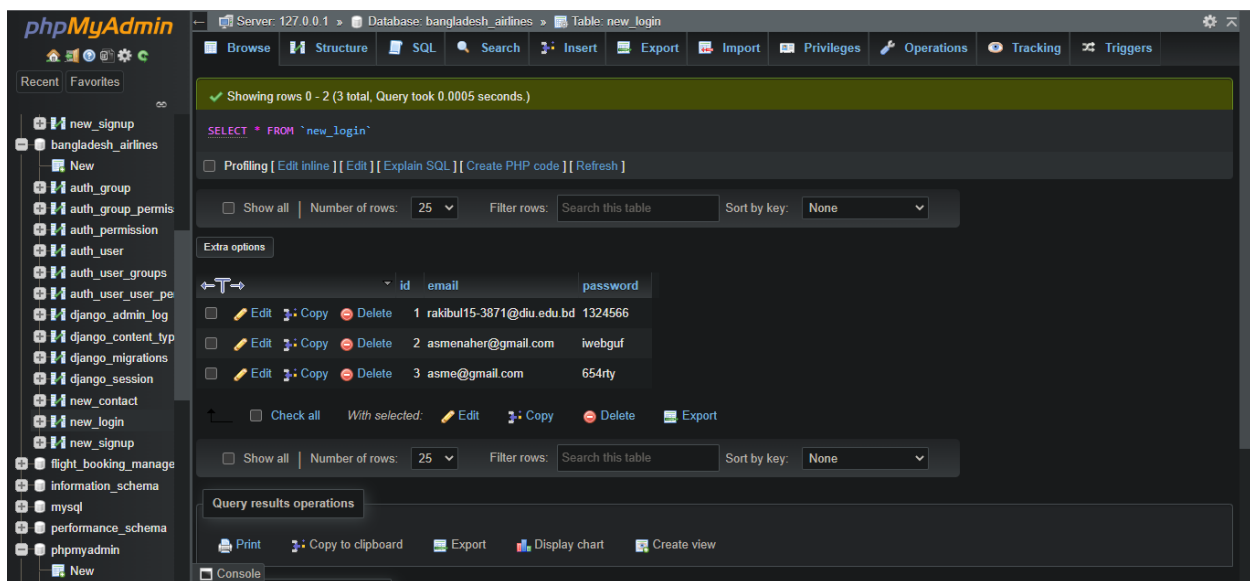
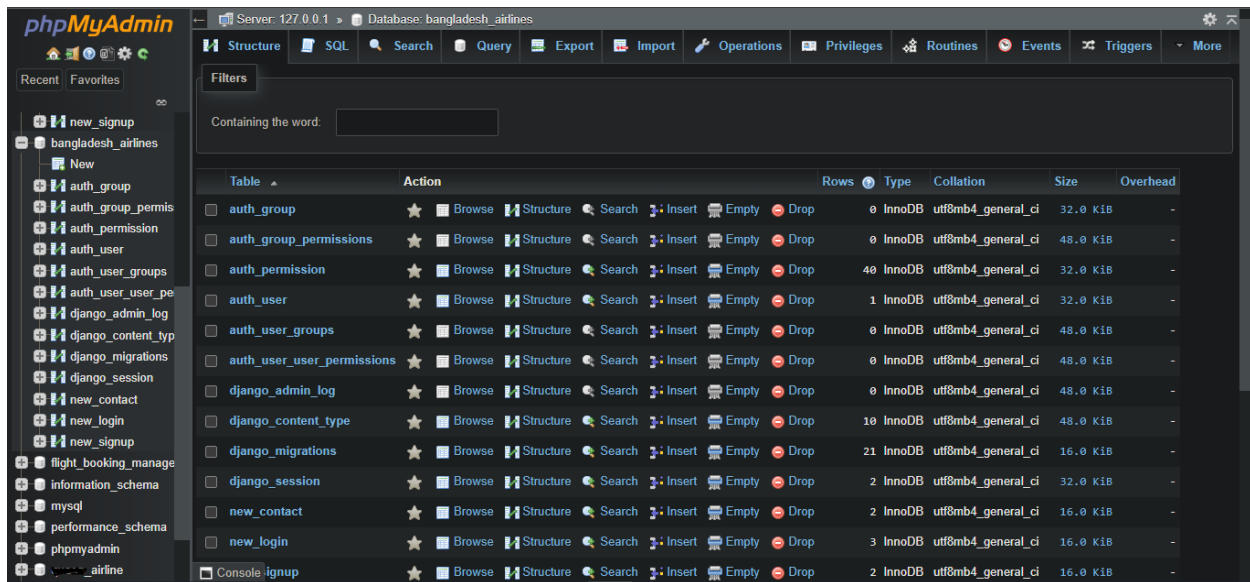
☐ asme@gmail.com

☐ asmenaher@gmail.com

☐ rakibul15-3871@diu.edu.bd

3 logins


# MySQL



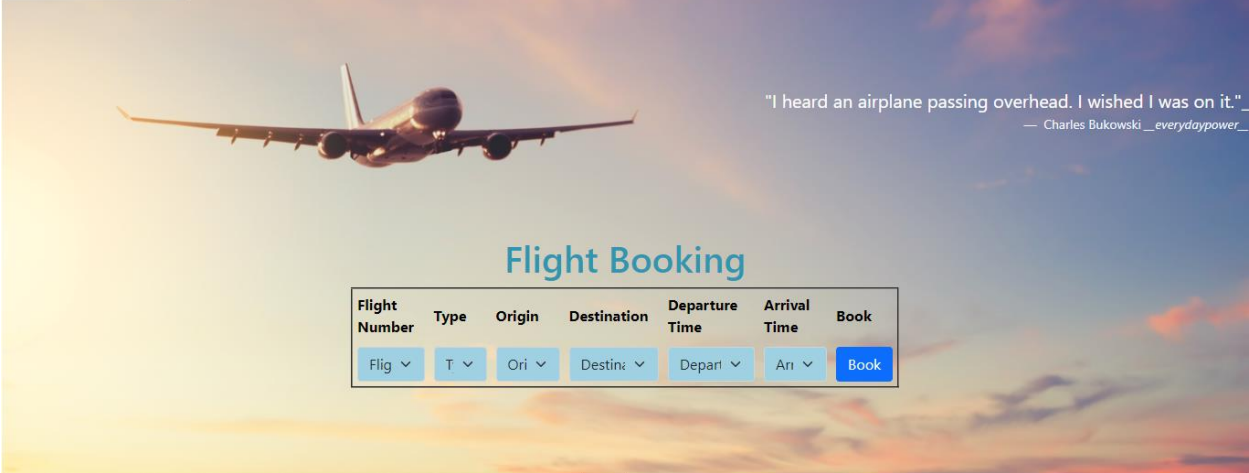
## Frontend

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](#) [About](#) [Contact](#) [Log in](#) [Sign up](#)


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

 [Home](#) [About](#) [Contact](#) [Log in](#) [Sign up](#)


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](#)

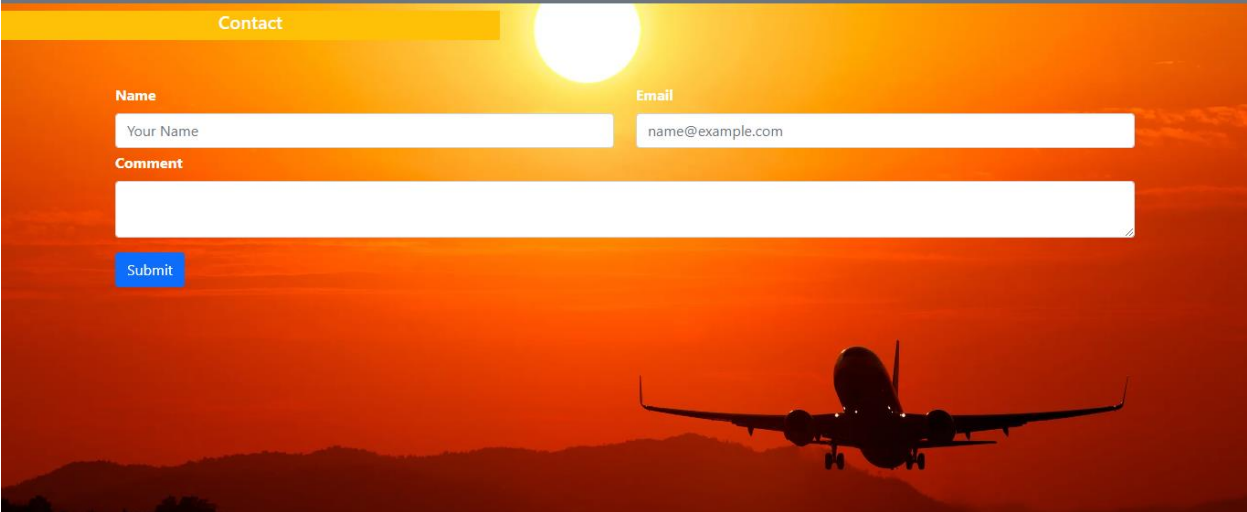
Contact


Name

Email

Comment

Submit

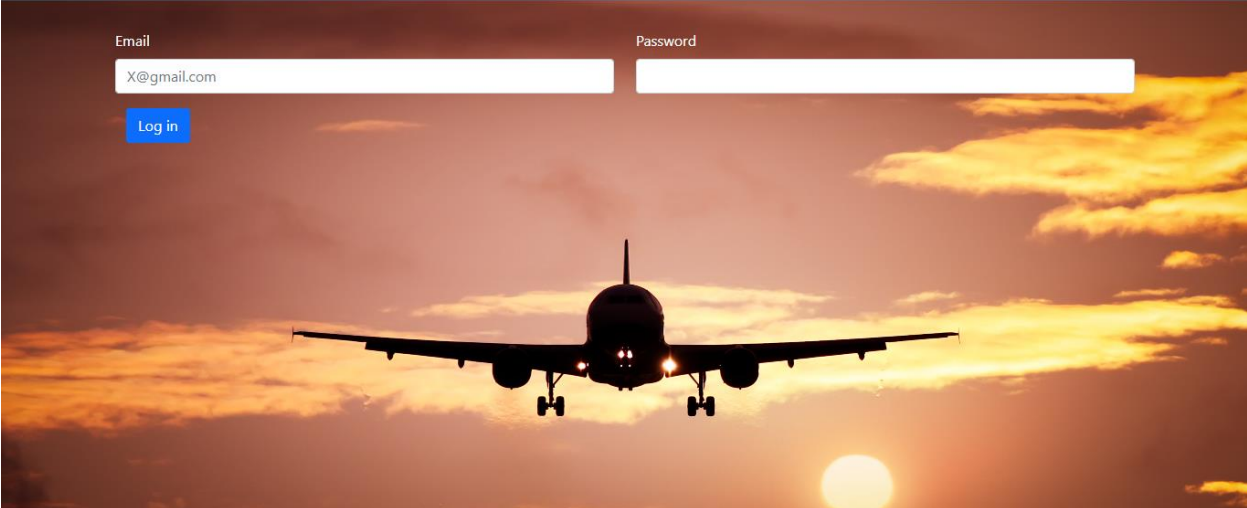



 [Home](#) [About](#) [Contact](#) [Log in](#) [Sign up](#)

Email

Password

Log in



[Home](#) [About](#) [Contact](#) [Log in](#) [Sign up](#)

First Name

Last Name

Email

Password

Present Address

Zip code

Country

Sign in



## 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](https://github.com/cryptic91/Bangladesh_Airlines)