

## Practice Project: Customer360



**Estimated time needed:** 45 minutes

This practice project will give you first hand experience applying the skills you learned in Django application development.

### Scenario

Your organization is consolidating different platforms and wants to store customer communication records in a central location. As a software engineer, you are tasked to develop this Customer360 application using Django.

A communication record stores Channel, Direction (inbound or outbound) and Summary.

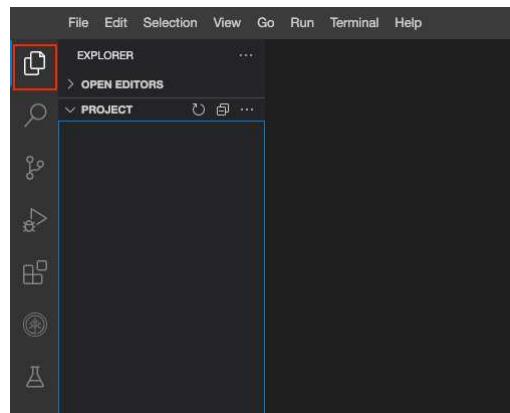
### Objectives

- Develop a screen to capture communication record
- Display interaction in last 30 days
- Provide professional customer management

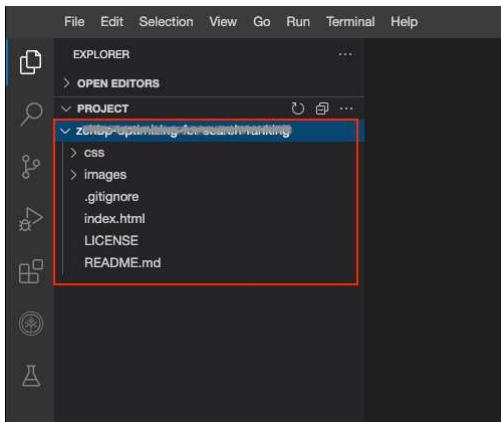
## Working with Files in Cloud IDE

If you are new to Cloud IDE, this section will show you how to create and edit files that are part of your project in Cloud IDE.

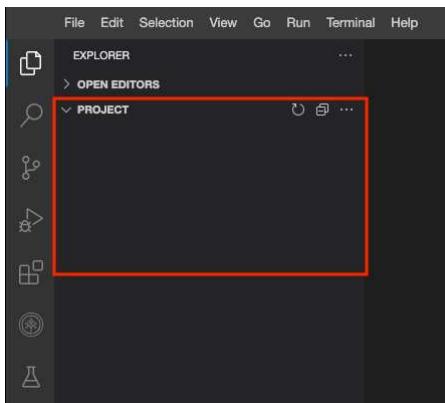
To view your files and directories inside Cloud IDE, click the file icon to reveal it.



If you have cloned (using the `git clone` command) boilerplate/starting code, then it will look like the image below:

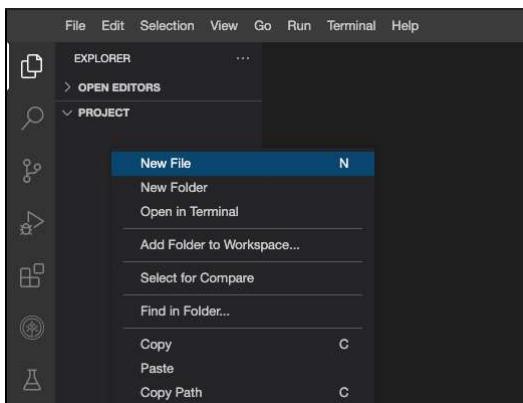


If you have not cloned and are starting with a blank project, it will look like this:

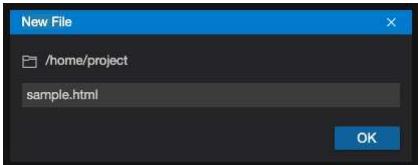


## Create a New File

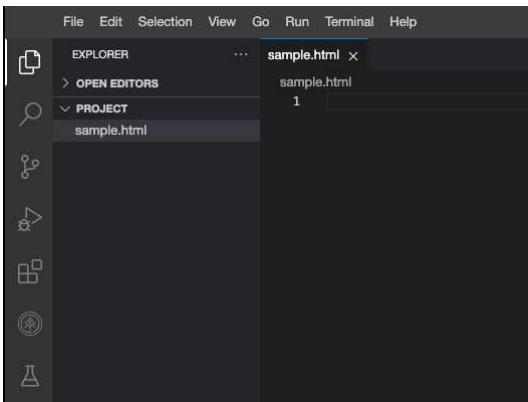
To create a new file in your project, right-click and select the **New File** option. You can also choose **File** -> **New File** to do the same.



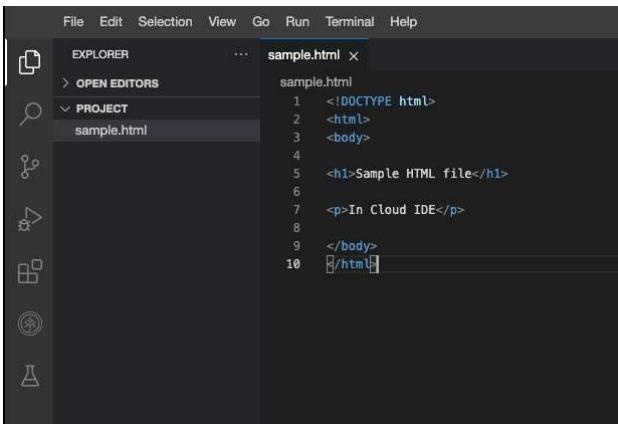
You will then be prompted to name the new file. In this scenario, let's name it `sample.html`.



Clicking the file name `sample.html` in the directory structure will open the file on the right pane. You can create all different types of files; for example, `FILE_NAME.js` for JavaScript files.

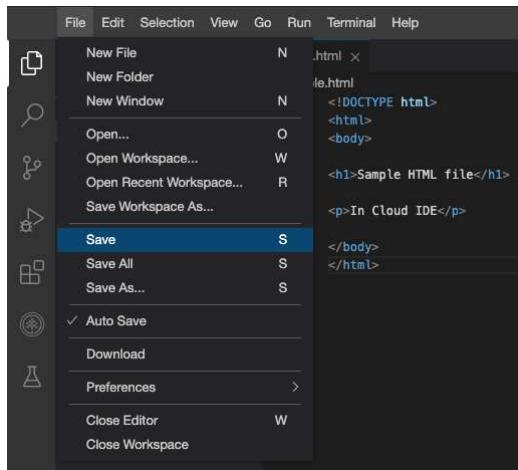


In the example below, we pasted some basic HTML code and then saved the file.



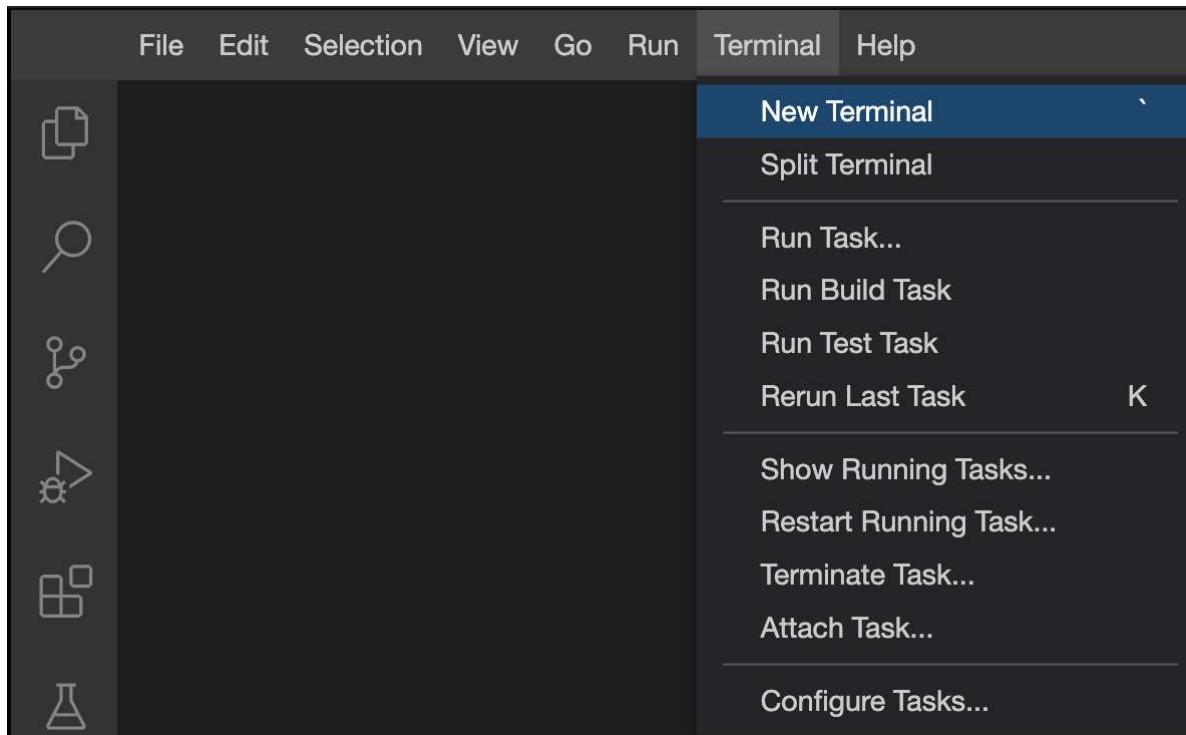
We save this file by:

- Going to the menu.
- Press Command + S on Mac or CTRL + S on Windows.
- Alternatively, it will Autosave your work as well.



## Set-up: Create a Django Application

1. Open a terminal window using the editor's menu: Select Terminal > New Terminal.



2. If you are not currently in the project folder, copy and paste the following code to change to your project folder. Select the copy button to the right of the code to copy it.

```
1. 1  
1. cd /home/project  
Copied! Executed!
```

3. Ensure pip is installed.

```
1. 1
1. python3.11 -m ensurepip
Copied! | Executed!
```

4. Install Django.

```
1. 1
1. python3.11 -m pip install Django
Copied! | Executed!
```

5. Create a project.

```
1. 1
1. django-admin startproject customer360
Copied! | Executed!
```

6. Change the directory so that it works in the lab.

```
1. 1
1. cd customer360
Copied! | Executed!
```

7. Run migration before running the application for the first time.

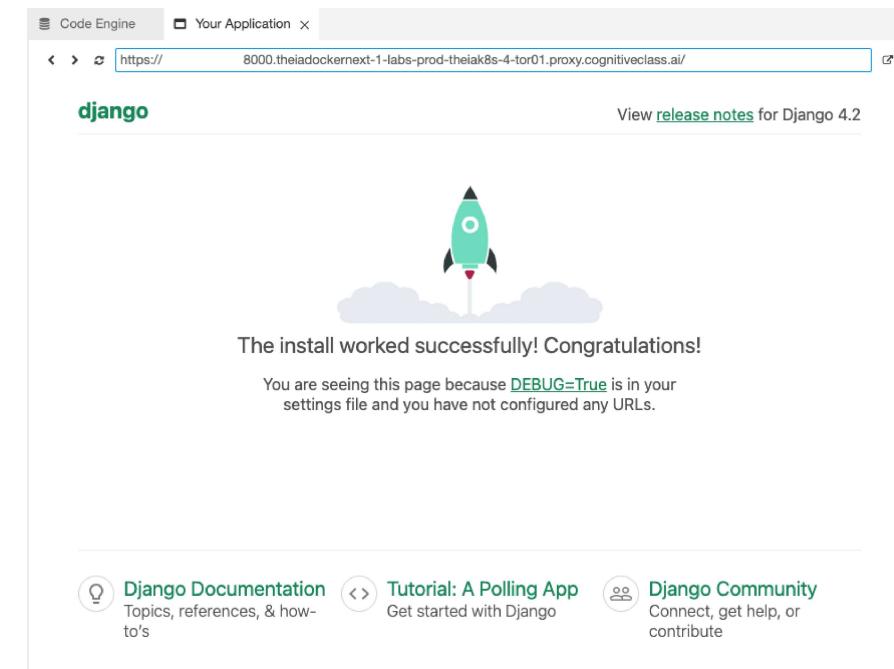
```
1. 1
1. python3.11 manage.py migrate
Copied! | Executed!
```

8. Run the server successfully this time.

```
1. 1
1. python3.11 manage.py runserver
Copied! | Executed!
```

[Launch Application](#)

9. It will look like the image below:



10. In your terminal, press CTRL+C to stop your web server.

## Task 1: Modify Settings

Now that your environment project is set up, you are set to start doing work.

Each application you write in Django consists of a Python package that follows a certain convention. Django comes with a utility that automatically generates the basic directory structure of an app, so you can focus on writing code rather than creating directories.

You will make changes to `settings.py` in the `customer360` app:

[Open `settings.py` in IDE](#)

### Allowed Hosts

A list of strings representing the host/domain names that this Django site can serve.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
1. ALLOWED_HOSTS=[ "*" ]
```

Copied!

### Installed Apps

A list of strings designating all applications that are enabled in this Django installation. Each string should be a dotted Python path to:

- an application configuration class (preferred), or
- a package containing an application.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
1. INSTALLED_APPS = [
2.     'django.contrib.admin',
3.     'django.contrib.auth',
4.     'django.contrib.contenttypes',
5.     'django.contrib.sessions',
6.     'django.contrib.messages',
7.     'django.contrib.staticfiles',
8.     'customer360'
9. ]
```

Copied!

### List of Trusted Origins

A list of trusted origins for unsafe requests (for example, POST).

For requests that include the Origin header, Django CSRF protection requires that the header match the origin present in the Host header.

```
1. 1
2. CSRF_TRUSTED_ORIGINS = [ 'https://*.cognitiveclass.ai' ]
```

Copied!

### Import OS

To be able to use `path` property. You need to import the `os` module. Add an `import` statement near the top of the file after `from pathlib...`

```
1. 1
2. 2
1. from pathlib import Path
2. import os
```

Copied!

### Configure Additional Static Files Directory

This setting defines the additional locations the staticfiles app will traverse if the `FileSystemFinder` finder is enabled, for example, if you use the `collectstatic` or `findstatic` management command or use the static file serving view.

For clarity, add it after `STATIC_URL`.

```
1. 1
2. 2
3. 3
```

```
1. STATICFILES_DIRS = (
2.     os.path.join(BASE_DIR, "static/"),
3. )
```

**Copied!**

You can see the complete file below.

► Completed settings.py

## Task 2: Create Models

Let's define our models that will help us store data and build the UI.

A model is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data you're storing. Generally, each model maps to a single database table.

### Define models

Start with creating a model file in `customer360/customer360/models.py`. Run the following script to create the file.

```
1. 1
2. touch /home/project/customer360/customer360/models.py
```

**Copied!** **Executed!****Open models.py in IDE**

And define the following:

#### Import

```
1. 1
2. from django.db import models
```

**Copied!**

#### Customer Model

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. class Customer(models.Model):
11.     id = models.AutoField(primary_key=True)
12.     name = models.CharField(max_length=100)
13.     email = models.EmailField(max_length=100)
14.     phone = models.CharField(max_length=20)
15.     address = models.CharField(max_length=200)
16.
17.     def __str__(self):
18.         return str(self.id)
```

**Copied!**

#### Interaction model

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19.
20. class Interaction(models.Model):
21.     CHANNEL_CHOICES = [
22.         ('phone', 'Phone'),
```

```
4.         ('sms', 'SMS'),
5.         ('email', 'Email'),
6.         ('letter', 'Letter'),
7.     ]
8.
9. DIRECTION_CHOICES = [
10.     ('inbound', 'Inbound'),
11.     ('outbound', 'Outbound'),
12. ]
13.
14. customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
15. channel = models.CharField(max_length=15, choices=CHANNEL_CHOICES)
16. direction = models.CharField(max_length=10, choices=DIRECTION_CHOICES)
17. interaction_date = models.DateField(auto_now_add=True)
18. summary = models.TextField()
```

Copied!

You can see the complete file below:

## ► Completed models.py

### **Task 3: Create Templates**

You now need to add a few HTML files that will show our models and let users interact with the application.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6

1. mkdini /home/project/customer360/customer360/templates
2. touch /home/project/customer360/customer360/templates/add.html
3. touch /home/project/customer360/customer360/templates/base.html
4. touch /home/project/customer360/customer360/templates/index.html
5. touch /home/project/customer360/customer360/templates/interact.html
6. touch /home/project/customer360/customer360/templates/summary.html
```

Copied!

You should review each HTML content as you paste the models into the relevant file.

## base.html

You will start with base template, aptly named as `base.html`. This file contains sections that we will fill in with other templates.

[Open base.html in IDE](#)

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26

1. {%- load static %}
2. <!DOCTYPE html>
3. <html>
4.     <head>
5.         <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6.         <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
7.         <link rel="stylesheet" href="{% static 'css/main.css' %}">
8.     </head>
9.     <body>
10.         <nav class="navbar navbar-default">
11.             <ul class="nav navbar-nav">
12.                 <li>
```

```

13.           <a style="color:black;" href="/">Home</a>
14.           </li>
15.           <li>
16.             <a style="color:black;" href="/create">New Customer</a>
17.           </li>
18.           <li>
19.             <a style="color:black;" href="/summary">Summary</a>
20.           </li>
21.         </ul>
22.       </nav>
23.       {% block content %}
24.     {% endblock %}
25.   </body>
26. </html>
```

Copied!

## index.html

Next you will fill the `index.html` with HTML content. This is our landing page.

[Open index.html in IDE](#)

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52

1. {% extends 'base.html' %}
2. {% load static %}
3. {% block content %}
4. <html>
5.   <head>
6.     <title>Home Page</title>
7.   </head>
8.   <script>
9.     function set_customer(){
10.       var cinput = document.querySelector('input[name="selected_customers"]:checked');
11.       if (cinput){
12.         cid = cinput.value;
13.         window.location = "/interact/"+cid;
14.       }
15.       else
16.         alert("Please select a customer");
17.     }

```

```

18. </script>
19. <body>
20.   <h1>Welcome to Customer 360</h1>
21.   <p>Interact and Manage your Customers</p>
22.   <a class="btn btn-primary" style="font-weight:bold; display:inline" onclick="set_customer()">Interact</a>
23.
24.   <table class="table">
25.     <thead>
26.       <tr>
27.         <th>Customer ID</th>
28.         <th>Name</th>
29.         <th>Email</th>
30.         <th>Phone</th>
31.         <th>Address</th>
32.         <th>Selected</th>
33.       </tr>
34.     </thead>
35.     <tbody>
36.       {% for customer in customers %}
37.         <tr>
38.           <td>{{customer.id }}</td>
39.           <td>{{customer.name }}</td>
40.           <td>{{customer.email }}</td>
41.           <td>{{customer.phone }}</td>
42.           <td>{{customer.address }}</td>
43.           <td>
44.             <input type="radio" name="selected_customers" value="{{ customer.id }}>
45.           </td>
46.         </tr>
47.       {% endfor %}
48.     </tbody>
49.   </table>
50. </body>
51. </html>
52. {% endblock content %}
```

Copied!

## add.html

You will now create the template for adding a new customer. You will notice that it extends to `base.html`.

Open add.html in IDE

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33

1. {% extends 'base.html' %}
2. {% load static %}
3. {% block content %}
4. <html>
5.   <head>
6.     <title>Add a Customer</title>
7.   </head>
8.   <body>
9.     <h1>Add a new Customer</h1>
10.    <form class="form" method="post" action="/create/">
11.      {% csrf_token %}
12.      <div class="form-group">
13.        <label for="Name">Name </label>
14.        <input type="text" name="name" required>
15.      </div>
```

```
16.      <div class="form-group">
17.        <label for="Email">Email </label>
18.        <input type="email" name="email" required>
19.      </div>
20.      <div class="form-group">
21.        <label for="Phone">Phone</label>
22.        <input type="tel" name="phone" required>
23.      </div>
24.      <div class="form-group">
25.        <label for="Address">Address</label>
26.        <input type="text" name="address" required>
27.      </div>
28.      <button type="submit" class="btn btn-success">Add</button>
29.      <p> {{ msg }} </p>
30.    </form>
31.  </body>
32. </html>
33. {% endblock content %}
```

Copied!

## interact.html

To record an interaction with a customer, we use the template `interact.html`.

[Open interact.html in IDE](#)

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
60. 60
61. 61

1. {% extends 'base.html' %}
2. {% load static %}
3. {% block content %}
4. <html>
```

```

5.      <head>
6.          <title>Interact & Manage</title>
7.      </head>
8.      <script>
9.          function selectButton(element) {
10.              var buttons = element.parentElement.getElementsByName("btn");
11.              for (var i = 0; i < buttons.length; i++) {
12.                  buttons[i].classList.remove("active");
13.              }
14.              element.classList.add("active");
15.          }
16.          function check_selected(){
17.              var dirinput = document.querySelector('input[name="direction"]:checked');
18.              var chaninput = document.querySelector('input[name="channel"]:checked');
19.              var summary = document.querySelector('textareaname="summary"]').value;
20.
21.              if (!dirinput || !chaninput || summary === ""){
22.                  alert("Please fill all required fields");
23.                  return false;
24.              }
25.              return true;
26.          }
27.      </script>
28.      <body>
29.          <h1>Interact With Your Customers</h1>
30.          <form class="form" method="post" onsubmit="return check_selected()" action="#">
31.              {% csrf_token %}
32.              <div class="form-group">
33.                  <label>Channel</label>
34.                  <div class="btn-group" data-toggle="buttons">
35.                      {% for channel in channels %}
36.                          <label class="btn btn-outline-primary" onclick="selectButton(this)">
37.                              <input type="radio" name="channel" value="{{ channel.0 }}" required> {{ channel.1 }}
38.                          </label>
39.                      {% endfor %}
40.                  </div>
41.              </div>
42.              <div class="form-group">
43.                  <label>Direction</label>
44.                  <div class="btn-group" data-toggle="buttons">
45.                      {% for direction in directions %}
46.                          <label class="btn btn-outline-primary" onclick="selectButton(this)">
47.                              <input type="radio" name="direction" value="{{ direction.0 }}" required> {{ direction.1 }}
48.                          </label>
49.                      {% endfor %}
50.                  </div>
51.              </div>
52.              <div class="form-group">
53.                  <label>Summary</label>
54.                  <textarea name="summary"></textarea>
55.              </div>
56.              <button type="submit" class="btn btn-success">Save Interaction</button>
57.              <p>{{ msg }}</p>
58.          </form>
59.      </body>
60.  </html>
61.  {% endblock content %}
```

Copied!

## summary.html

Finally, you will fill the HTML template `summary.html` as follows:

[Open `summary.html` in IDE](#)

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
```

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
60. 60
61. 61
62. 62
63. 63
64. 64
65. 65
66. 66
67. 67
68. 68
69. 69
70. 70
71. 71
72. 72
73. 73
74. 74
75. 75
76. 76
77. 77
78. 78
79. 79
80. 80
81. 81
82. 82
83. 83
84. 84
85. 85
86. 86
87. 87
88. 88
89. 89
90. 90
91. 91
92. 92
93. 93
94. 94
95. 95
96. 96
97. 97
98. 98
99. 99
100. 100
101. 101
102. 102
103. 103
104. 104
105. 105
106. 106
107. 107
108. 108
109. 109
110. 110
111. 111
112. 112
113. 113
114. 114
115. 115
116. 116
117. 117
118. 118
119. 119
120. 120
121. 121
122. 122
123. 123
124. 124
125. 125
126. 126
127. 127
128. 128
129. 129
130. 130
131. 131
132. 132
133. 133
134. 134
135. 135
136. 136
137. 137
138. 138
139. 139
140. 140
141. 141
142. 142
143. 143
144. 144
145. 145
146. 146
147. 147
148. 148
149. 149
150. 150
151. 151
152. 152
153. 153
154. 154
155. 155
156. 156
157. 157
158. 158
159. 159
160. 160
161. 161
162. 162
163. 163
164. 164
165. 165
166. 166
167. 167
168. 168
169. 169
170. 170
171. 171
172. 172
173. 173
174. 174
175. 175
176. 176
177. 177
178. 178
179. 179
180. 180
181. 181
182. 182
183. 183
184. 184
185. 185
186. 186
187. 187
188. 188
189. 189
190. 190
191. 191
192. 192
193. 193
194. 194
195. 195
196. 196
197. 197
198. 198
199. 199
200. 200
201. 201
202. 202
203. 203
204. 204
205. 205
206. 206
207. 207
208. 208
209. 209
210. 210
211. 211
212. 212
213. 213
214. 214
215. 215
216. 216
217. 217
218. 218
219. 219
220. 220
221. 221
222. 222
223. 223
224. 224
225. 225
226. 226
227. 227
228. 228
229. 229
230. 230
231. 231
232. 232
233. 233
234. 234
235. 235
236. 236
237. 237
238. 238
239. 239
240. 240
241. 241
242. 242
243. 243
244. 244
245. 245
246. 246
247. 247
248. 248
249. 249
250. 250
251. 251
252. 252
253. 253
254. 254
255. 255
256. 256
257. 257
258. 258
259. 259
260. 260
261. 261
262. 262
263. 263
264. 264
265. 265
266. 266
267. 267
268. 268
269. 269
270. 270
271. 271
272. 272
273. 273
274. 274
275. 275
276. 276
277. 277
278. 278
279. 279
280. 280
281. 281
282. 282
283. 283
284. 284
285. 285
286. 286
287. 287
288. 288
289. 289
290. 290
291. 291
292. 292
293. 293
294. 294
295. 295
296. 296
297. 297
298. 298
299. 299
300. 300
301. 301
302. 302
303. 303
304. 304
305. 305
306. 306
307. 307
308. 308
309. 309
310. 310
311. 311
312. 312
313. 313
314. 314
315. 315
316. 316
317. 317
318. 318
319. 319
320. 320
321. 321
322. 322
323. 323
324. 324
325. 325
326. 326
327. 327
328. 328
329. 329
330. 330
331. 331
332. 332
333. 333
334. 334
335. 335
336. 336
337. 337
338. 338
339. 339
340. 340
341. 341
342. 342
343. 343
344. 344
345. 345
346. 346
347. 347
348. 348
349. 349
350. 350
351. 351
352. 352
353. 353
354. 354
355. 355
356. 356
357. 357
358. 358
359. 359
360. 360
361. 361
362. 362
363. 363
364. 364
365. 365
366. 366
367. 367
368. 368
369. 369
370. 370
371. 371
372. 372
373. 373
374. 374
375. 375
376. 376
377. 377
378. 378
379. 379
380. 380
381. 381
382. 382
383. 383
384. 384
385. 385
386. 386
387. 387
388. 388
389. 389
390. 390
391. 391
392. 392
393. 393
394. 394
395. 395
396. 396
397. 397
398. 398
399. 399
400. 400
401. 401
402. 402
403. 403
404. 404
405. 405
406. 406
407. 407
408. 408
409. 409
410. 410
411. 411
412. 412
413. 413
414. 414
415. 415
416. 416
417. 417
418. 418
419. 419
420. 420
421. 421
422. 422
423. 423
424. 424
425. 425
426. 426
427. 427
428. 428
429. 429
430. 430
431. 431
432. 432
433. 433
434. 434
435. 435
436. 436
437. 437
438. 438
439. 439
440. 440
441. 441
442. 442
443. 443
444. 444
445. 445
446. 446
447. 447
448. 448
449. 449
450. 450
451. 451
452. 452
453. 453
454. 454
455. 455
456. 456
457. 457
458. 458
459. 459
460. 460
461. 461
462. 462
463. 463
464. 464
465. 465
466. 466
467. 467
468. 468
469. 469
470. 470
471. 471
472. 472
473. 473
474. 474
475. 475
476. 476
477. 477
478. 478
479. 479
480. 480
481. 481
482. 482
483. 483
484. 484
485. 485
486. 486
487. 487
488. 488
489. 489
490. 490
491. 491
492. 492
493. 493
494. 494
495. 495
496. 496
497. 497
498. 498
499. 499
500. 500
501. 501
502. 502
503. 503
504. 504
505. 505
506. 506
507. 507
508. 508
509. 509
510. 510
511. 511
512. 512
513. 513
514. 514
515. 515
516. 516
517. 517
518. 518
519. 519
520. 520
521. 521
522. 522
523. 523
524. 524
525. 525
526. 526
527. 527
528. 528
529. 529
530. 530
531. 531
532. 532
533. 533
534. 534
535. 535
536. 536
537. 537
538. 538
539. 539
540. 540
541. 541
542. 542
543. 543
544. 544
545. 545
546. 546
547. 547
548. 548
549. 549
550. 550
551. 551
552. 552
553. 553
554. 554
555. 555
556. 556
557. 557
558. 558
559. 559
560. 560
561. 561
562. 562
563. 563
564. 564
565. 565
566. 566
567. 567
568. 568
569. 569
570. 570
571. 571
572. 572
573. 573
574. 574
575. 575
576. 576
577. 577
578. 578
579. 579
580. 580
581. 581
582. 582
583. 583
584. 584
585. 585
586. 586
587. 587
588. 588
589. 589
590. 590
591. 591
592. 592
593. 593
594. 594
595. 595
596. 596
597. 597
598. 598
599. 599
600. 600
601. 601
602. 602
603. 603
604. 604
605. 605
606. 606
607. 607
608. 608
609. 609
610. 610
611. 611
612. 612
613. 613
614. 614
615. 615
616. 616
617. 617
618. 618
619. 619
620. 620
621. 621
622. 622
623. 623
624. 624
625. 625
626. 626
627. 627
628. 628
629. 629
630. 630
631. 631
632. 632
633. 633
634. 634
635. 635
636. 636
637. 637
638. 638
639. 639
640. 640
641. 641
642. 642
643. 643
644. 644
645. 645
646. 646
647. 647
648. 648
649. 649
650. 650
651. 651
652. 652
653. 653
654. 654
655. 655
656. 656
657. 657
658. 658
659. 659
660. 660
661. 661
662. 662
663. 663
664. 664
665. 665
666. 666
667. 667
668. 668
669. 669
670. 670
671. 671
672. 672
673. 673
674. 674
675. 675
676. 676
677. 677
678. 678
679. 679
680. 680
681. 681
682. 682
683. 683
684. 684
685. 685
686. 686
687. 687
688. 688
689. 689
690. 690
691. 691
692. 692
693. 693
694. 694
695. 695
696. 696
697. 697
698. 698
699. 699
700. 700
701. 701
702. 702
703. 703
704. 704
705. 705
706. 706
707. 707
708. 708
709. 709
710. 710
711. 711
712. 712
713. 713
714. 714
715. 715
716. 716
717. 717
718. 718
719. 719
720. 720
721. 721
722. 722
723. 723
724. 724
725. 725
726. 726
727. 727
728. 728
729. 729
730. 730
731. 731
732. 732
733. 733
734. 734
735. 735
736. 736
737. 737
738. 738
739. 739
740. 740
741. 741
742. 742
743. 743
744. 744
745. 745
746. 746
747. 747
748. 748
749. 749
750. 750
751. 751
752. 752
753. 753
754. 754
755. 755
756. 756
757. 757
758. 758
759. 759
760. 760
761. 761
762. 762
763. 763
764. 764
765. 765
766. 766
767. 767
768. 768
769. 769
770. 770
771. 771
772. 772
773. 773
774. 774
775. 775
776. 776
777. 777
778. 778
779. 779
780. 780
781. 781
782. 782
783. 783
784. 784
785. 785
786. 786
787. 787
788. 788
789. 789
790. 790
791. 791
792. 792
793. 793
794. 794
795. 795
796. 796
797. 797
798. 798
799. 799
800. 800
801. 801
802. 802
803. 803
804. 804
805. 805
806. 806
807. 807
808. 808
809. 809
810. 810
811. 811
812. 812
813. 813
814. 814
815. 815
816. 816
817. 817
818. 818
819. 819
820. 820
821. 821
822. 822
823. 823
824. 824
825. 825
826. 826
827. 827
828. 828
829. 829
830. 830
831. 831
832. 832
833. 833
834. 834
835. 835
836. 836
837. 837
838. 838
839. 839
840. 840
841. 841
842. 842
843. 843
844. 844
845. 845
846. 846
847. 847
848. 848
849. 849
850. 850
851. 851
852. 852
853. 853
854. 854
855. 855
856. 856
857. 857
858. 858
859. 859
860. 860
861. 861
862. 862
863. 863
864. 864
865. 865
866. 866
867. 867
868. 868
869. 869
870. 870
871. 871
872. 872
873. 873
874. 874
875. 875
876. 876
877. 877
878. 878
879. 879
880. 880
881. 881
882. 882
883. 883
884. 884
885. 885
886. 886
887. 887
888. 888
889. 889
890. 890
891. 891
892. 892
893. 893
894. 894
895. 895
896. 896
897. 897
898. 898
899. 899
900. 900
901. 901
902. 902
903. 903
904. 904
905. 905
906. 906
907. 907
908. 908
909. 909
910. 910
911. 911
912. 912
913. 913
914. 914
915. 915
916. 916
917. 917
918. 918
919. 919
920. 920
921. 921
922. 922
923. 923
924. 924
925. 925
926. 926
927. 927
928. 928
929. 929
930. 930
931. 931
932. 932
933. 933
934. 934
935. 935
936. 936
937. 937
938. 938
939. 939
940. 940
941. 941
942. 942
943. 943
944. 944
945. 945
946. 946
947. 947
948. 948
949. 949
950. 950
951. 951
952. 952
953. 953
954. 954
955. 955
956. 956
957. 957
958. 958
959. 959
960. 960
961. 961
962. 962
963. 963
964. 964
965. 965
966. 966
967. 967
968. 968
969. 969
970. 970
971. 971
972. 972
973. 973
974. 974
975. 975
976. 976
977. 977
978. 978
979. 979
980. 980
981. 981
982. 982
983. 983
984. 984
985. 985
986. 986
987. 987
988. 988
989. 989
990. 990
991. 991
992. 992
993. 993
994. 994
995. 995
996. 996
997. 997
998. 998
999. 999
1000. 1000
1001. 1001
1002. 1002
1003. 1003
1004. 1004
1005. 1005
1006. 1006
1007. 1007
1008. 1008
1009. 1009
1010. 1010
1011. 1011
1012. 1012
1013. 1013
1014. 1014
1015. 1015
1016. 1016
1017. 1017
1018. 1018
1019. 1019
1020. 1020
1021. 1021
1022. 1022
1023. 1023
1024. 1024
1025. 1025
1026. 1026
1027. 1027
1028. 1028
1029. 1029
1030. 1030
1031. 1031
1032. 1032
1033. 1033
1034. 1034
1035. 1035
1036. 1036
1037. 1037
1038. 1038
1039. 1039
1040. 1040
1041. 1041
1042. 1042
1043. 1043
1044. 1044
1045. 1045
1046. 1046
1047. 1047
1048. 1048
1049. 1049
1050. 1050
1051. 1051
1052. 1052
1053. 1053
1054. 1054
1055. 1055
1056. 1056
1057. 1057
1058. 1058
1059. 1059
1060. 1060
1061. 1061
1062. 1062
1063. 1063
1064. 1064
1065. 1065
1066. 1066
1067. 1067
1068. 1068
1069. 1069
1070. 1070
1071. 1071
1072. 1072
1073. 1073
1074. 1074
1075. 1075
1076. 1076
1077. 1077
1078. 1078
1079. 1079
1080. 1080
1081. 1081
1082. 1082
1083. 1083
1084. 1084
1085. 1085
1086. 1086
1087. 1087
1088. 1088
1089. 1089
1090. 1090
1091. 1091
1092. 1092
1093. 1093
1094. 1094
1095. 1095
1096. 1096
1097. 1097
1098. 1098
1099. 1099
1100. 1100
1101. 1101
1102. 1102
1103. 1103
1104. 1104
1105. 1105
1106. 1106
1107. 1107
1108. 1108
1109. 1109
1110. 1110
1111. 1111
1112. 1112
1113. 1113
1114. 1114
1115. 1115
1116. 1116
1117. 1117
1118. 1118
1119. 1119
1120. 1120
1121. 1121
1122. 1122
1123. 1123
1124. 1124
1125. 1125
1126. 1126
1127. 1127
1128. 1128
1129. 1129
1130. 1130
1131. 1131
1132. 1132
1133. 1133
1134. 1134
1135. 1135
1136. 1136
1137. 1137
1138. 1138
1139. 1139
1140. 1140
1141. 1141
1142. 1142
1143. 1143
1144. 1144
1145. 1145
1146. 1146
1147. 1147
1148. 1148
1149. 1149
1150. 1150
1151. 1151
1152. 1152
1153. 1153
1154. 1154
1155. 1155
1156. 1156
1157. 1157
1158. 1158
1159. 1159
1160. 1160
1161. 1161
1162. 1162
1163. 1163
1164. 1164
1165. 1165
1166. 1166
1167. 1167
1168. 1168
1169. 1169
1170. 1170
1171. 1171
1172. 1172
1173. 1173
1174. 1174
1175. 1175
1176. 1176
1177. 1177
1178. 1178
1179. 1179
1180. 1180
1181. 1181
1182. 1182
1183. 1183
1184. 1184
1185. 1185
1186. 1186
1187. 1187
1188. 1188
1189. 1189
1190. 1190
1191. 1191
1192. 1192
1193. 1193
1194. 1194
1195. 1195
1196. 1196
1197. 1197
1198. 1198
1199. 1199
1200. 1200
1201. 1201
1202. 1202
1203. 1203
1204. 1204
1205. 1205
1206. 1206
1207. 1207
1208. 1208
1209. 12
```

```

5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11

1. def create_customer(request):
2.     if request.method == "POST":
3.         name = request.POST["name"]
4.         email = request.POST["email"]
5.         phone = request.POST["phone"]
6.         address = request.POST["address"]
7.         customer = Customer.objects.create(name=name, email=email, phone=phone, address=address)
8.         customer.save()
9.         msg = "Successfully Saved a Customer"
10.        return render(request, "add.html", context={"msg":msg})
11.    return render(request, "add.html")

```

Copied!

## Define Summary View

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12

1. def summary(request):
2.     thirty_days_ago = date.today() - timedelta(days=30)
3.     interactions = Interaction.objects.filter(interaction_date__gte=thirty_days_ago)
4.
5.     count = len(interactions)
6.     interactions = interactions.values("channel", "direction").annotate(count=Count('channel'))
7.     context = {
8.         "interactions": interactions,
9.         "count": count
10.    }
11.
12.    return render(request, "summary.html", context=context)

```

Copied!

## Define Interaction View

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22

1. def interact(request, cid):
2.
3.     channels = Interaction.CHANNEL_CHOICES
4.     directions = Interaction.DIRECTION_CHOICES
5.     context = {"channels": channels, "directions": directions}
6.
7.     if request.method == "POST":
8.
9.         customer = Customer.objects.get(id=cid)
10.        channel = request.POST["channel"]
11.        direction = request.POST["direction"]
12.        summary = request.POST["summary"]
13.        interaction = Interaction.objects.create(
14.            customer=customer,
15.            channel=channel,

```

```
16.         direction=direction,
17.         summary=summary)
18.     interaction.save()
19.     context["msg"] = "Interaction Success"
20.     return render(request,"interact.html",context=context)
21.
22. return render(request,"interact.html",context=context)
```

Copied!

## Completed File

You can see the completed file below.

### ▼ Completed views.py

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
```

```
1. from django.shortcuts import render
2. from datetime import date, timedelta
3. from django.db.models import Count
4. from .models import *
5.
6. # Create your views here.
7. def index(request):
8.     customers = Customer.objects.all()
9.     context = {"customers":customers}
10.    return render(request,"index.html",context=context)
11.
12. def create_customer(request):
13.     if request.method == "POST":
14.         name = request.POST["name"]
15.         email = request.POST["email"]
16.         phone = request.POST["phone"]
17.         address = request.POST["address"]
```

```

18.     customer = Customer.objects.create(name=name,email=email,phone=phone,address=address)
19.     customer.save()
20.     msg = "Successfully Saved a Customer"
21.     return render(request,"add.html",context={"msg":msg})
22. return render(request,"add.html")
23.
24.
25. def summary(request):
26.     thirty_days_ago = date.today() - timedelta(days=30)
27.     interactions = Interaction.objects.filter(interaction_date__gte=thirty_days_ago)
28.
29.     count = len(interactions)
30.     interactions = interactions.values("channel","direction").annotate(count=Count('channel'))
31.     context={
32.         "interactions":interactions,
33.         "count":count
34.     }
35.
36.     return render(request,"summary.html",context=context)
37.
38. def interact(request,cid):
39.
40.     channels = Interaction.CHANNEL_CHOICES
41.     directions = Interaction.DIRECTION_CHOICES
42.     context = {"channels":channels,"directions":directions}
43.
44.     if request.method == "POST":
45.
46.         customer = Customer.objects.get(id=cid)
47.         channel = request.POST["channel"]
48.         direction = request.POST["direction"]
49.         summary = request.POST["summary"]
50.         interaction = Interaction.objects.create(
51.             customer=customer,
52.             channel=channel,
53.             direction=direction,
54.             summary=summary)
55.         interaction.save()
56.         context["msg"] = "Interaction Success"
57.         return render(request,"interact.html",context=context)
58.
59.     return render(request,"interact.html",context=context)

```

**Copied!**

## Task 5: Create URLs

A clean, elegant URL scheme is an important detail in a high-quality web application. Django lets you design URLs however you want, with no framework limitations.

You will make changes to `urls.py` in the `customer360` app:

**Open urls.py in IDE**

### Import Views

```

1. 1
1. from . import views

```

**Copied!**

### Add URL Patterns

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
1. urlpatterns = [
2.     path('admin/', admin.site.urls),
3.     path('',views.index,name="index"),
4.     path('create/',views.create_customer,name='create_customer'),
5.     path('interact/<int:cid>',views.interact,name='interact'),
6.     path('summary/',views.summary,name='summary'),
7. ]

```

**Copied!**

You can see the complete file below.

### ▼ Completed urls.py

```

1. 1
2. 2
3. 3

```

```
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11

1. from django.contrib import admin
2. from django.urls import path
3. from . import views
4.
5. urlpatterns = [
6.     path('admin/', admin.site.urls),
7.     path('',views.index,name="index"),
8.     path('create/',views.create_customer,name='create_customer'),
9.     path('interact/<int:cids>',views.interact,name='interact'),
10.    path('summary/',views.summary,name='summary'),
11. ]
```

Copied!

## Task 6: Add Styling

To visually enhance the user experience, you are given the following CSS which you should add to your application.

```
1. 1
2. 2

1. mkdir -p /home/project/customer360/static/css
2. touch /home/project/customer360/static/css/main.css
```

Copied!

Executed!

Open main.css in IDE

### CSS content

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
```

```
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
60. 60
61. 61
62. 62
63. 63
64. 64
65. 65
66. 66
67. 67
68. 68
69. 69
70. 70
71. 71

1. body{
2.   text-align: center;
3.   margin: auto;
4. }
5. h1,p{
6.   font-weight: bold;
7. }
8. table{
9.   margin: auto;
10.  border: 2px solid #000000;
11.  margin-top: 50px;
12.  border-radius: 15px;
13.  background-color: white;
14.  text-align: center;
15. }
16. th{
17.   text-align: center;
18. }
19.
20. .nav{
21.   width: 200%;
22.   border: 2px solid black;
23.   border-radius: 10px;
24. }
25. .nav li{
26.   margin-right: 6%;
27.   margin-left: 6%;
28.   font-size: large;
29.   font-weight: bold;
30. }
31. .form{
32.   margin: auto;
33.   margin-top: 20px;
34.   background-color: white;
35.   text-align: center;
36.   width: 50%;
37.   height: 50px;
38.   border: 2px solid black;
39.   border-radius: 10px;
40. }
41. .form-group{
42.   display: flex;
43.   flex-direction: column;
44.   align-items: flex-start;
45.
46.   width:100%;
47.   padding: 10px;
48.   text-align: left;
49.   margin-top: 10px;
50.   width: 100%;
51. }
52. input[type="text"],input[type="email"],input[type="tel"]{
53.   width: 100%;
54. }
55. textarea{
56.   width: 500px;
57.   height: 200px;
58. }
59. .btn{
60.   display: flex;
61.   margin-top: 10px;
62.   margin-bottom: 10px;
63.   padding-left: 50px;
64.   padding-right: 50px;
65.   text-align: left;
66.   margin-left: 10px;
67. }
68. .active {
69.   background-color: #007bff;
70.   color: #fff;
71. }
```

Copied!

## Task 7: Run the Application

```

1. 1
2. 2
3. 3
4. 4

1. cd /home/project/customer360
2. python3.11 manage.py makemigrations customer360
3. python3.11 manage.py migrate
4. python3.11 manage.py runserver

```

Copied! Executed!

[Launch Customer360](#)

### Launch

When you launch the Customer360 app you will see a similar view as shown below.

The screenshot shows a web browser window with the URL <https://-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/>. The page title is "Welcome to Customer 360". Below the title, it says "Interact and Manage your Customers" and has a blue "Interact" button. At the top, there are three navigation links: "Home", "New Customer", and "Summary". A table below the header lists columns: Customer ID, Name, Email, Phone, Address, and Selected. There is one row of data in the table.

Customer ID	Name	Email	Phone	Address	Selected

### New Customer

There are no customer records because you are running it for the first time. Let's add one.

#### Add a new Customer

The form is titled "Add a new Customer". It contains four input fields: "Name" (with value "Joe Bloggs"), "Email" (with value "joe@bloggs.com"), "Phone" (with value "830-244-6127"), and "Address" (with value "4434 Morris Street San Antonio, TX 78205"). Below the form is a green "Add" button.

### List of Customers

You can now see your newly added customer on the main page.

The screenshot shows the Customer 360 application interface. At the top, there are three navigation tabs: "Home", "New Customer", and "Summary". Below the tabs, a large heading says "Welcome to Customer 360" with the subtext "Interact and Manage your Customers". A blue "Interact" button is centered below the subtext. The main content area displays a table with one customer record:

Customer ID	Name	Email	Phone	Address	Selected
1	Joe Bloggs	joe@bloggs.com	830-244-6127	4434 Morris Street San Antonio, TX 78205	<input type="radio"/>

### New Interaction record

You will now record a new interaction for this customer, by selecting the customer record and then **Interact**.

Then fill in the interaction details.

### Interact With Your Customers

The form is titled "Interact With Your Customers". It has three sections: "Channel", "Direction", and "Summary".

**Channel:** Buttons for Phone (selected), SMS, Email, and Letter.

**Direction:** Buttons for Inbound and Outbound (selected).

**Summary:** A text area containing the placeholder text "Called the customer to provide information about recent visit." A "Save Interaction" button is at the bottom.

### Summary

You can now go to the Summary page.

## Interactions in last 30 Days

Channel	Direction	Count
phone	outbound	1

Total : 1

## Task 8: Modifications (Optional)

Your challenge now is to enhance the Customer360 app and add some functionality. You can discuss these modifications with your peers.

1. Add a new field to customer model
2. Create a new interaction channel

### Add social media Field to Customer

#### Model change

In the `Customer` model, add a new optional field called `social_media`. And display it with customer details.

▼ Hint

```
1. 1
2. new_field = models.data_type(args)
```

Copied!

▼ Solution

```
1. 1
2. social_media = models.CharField(max_length=100, blank=True)
```

Copied!

#### Template change

Now you need to ensure that the new field `social_media` is added to the customer create form.

▼ Hint

```
1. 1
2. 2
3. 3
4. 4

1. <div class="form-group">
2.   <label for="Field">Field Label </label>
3.   <input type="Type" name="Name">
4. </div>
```

Copied!

▼ Solution

```
1. 1
2. 2
3. 3
4. 4

1. <div class="form-group">
2.   <label for="SocialMedia">Social Media </label>
3.   <input type="text" name="social_media">
4. </div>
```

Copied!

And that the `social_media` field is added in the grid on the landing page.

▼ Hint

```
1. 1
2. 2
3. 3
```

```
1. <th>Field Name</th>
2. ...
3. <td>{{mode.field_name }}</td>
```

**Copied!**  
▼ Solution

```
1. 1
2. 2
3. 3

1. <th>Social Media</th>
2. ...
3. <td>{{customer.social_media }}</td>
```

**Copied!**

## View change

1. Add the code for posting as a Social media text

▼ Hint

Use the `request.POST` method

▼ Solution

```
1. 1
1. social_media=request.POST["social_media"]
```

**Copied!**

2. Create a new Customer object using the `Customer.objects.create()` method

▼ Hint

Add the `social_media` attribute to the existing `Customer.objects.create()` method

▼ Solution

```
1. 1
1. customer = Customer.objects.create(name=name,email=email,phone=phone,address=address,social_media=social_media)
```

**Copied!**

## Add new interaction channel

This change is only in one place. Can you figure out where this should be?

▼ Hint

In Models file, in the Channel Choices array, add a new entry. The reason this change is only in one place is because in `interact.html`, we are iterating over possible channels instead of hard coding.

```
1. 1
1. ('new_choice', 'New Choice'),
```

**Copied!**

▼ Solution

```
1. 1
1. ('social_media', 'Social Media'),
```

**Copied!**

## Test your changes

To apply these model changes, you need to run certain procedures. These procedures will apply the model changes to your database.

▼ Hint

```
1. 1
2. 2
3. 3
```

```
1. # create migrations
2. # migrate
3. # run server to test
```

**Copied!**

▼ Solution

```
1. 1
```

2. 2  
3. 3

1. python3.11 manage.py makemigrations customer360  
2. python3.11 manage.py migrate  
3. python3.11 manage.py runserver

Copied!

[Launch Customer360](#)

## Summary

### Congratulations!

You have completed this practice project exercising the skills you learned in Django application development.

### Author(s)

- [Muhammad Yahya](#)

### Changelog

Date	Version	Changed by	Change Description
2023-08-07	0.1	Muhammad Yahya	Initial commit
2023-08-25	0.2	Ritika Joshi	Instructions updated based on beta testing

© IBM Corporation 2023. All rights reserved.