



# Phase 4: Transition

---

## 4.02 Build

### Version 7.0

#### Revision History

Date	Version	Description	Author
04-10-2023	7.0	Updated Reports Section	Cathy Zhao
04-10-2023	6.0	Updated Login and User Permissions	Anna Song
04-07-2023	5.0	Updated Claims Section	Jacquelyn Tiffin
04-01-2023	4.0	Added Reports Section	Megan Kwan
04-01-2023	3.0	Added Claims Section	Jacquelyn Tiffin
04-01-2023	2.0	Added Sales Section	Matthew Pate
04-01-2023	1.0	Initial Build Developed including Login and User Permissions	Anna Song

## Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Scope	4
1.3 References	4
<b>2. Login and User Permissions</b>	<b>4</b>
2.1 Login Web Page	4
2.1.1 Login Portal	4
2.1.2 Register Page	5
2.1.3 Account Creation	5
2.1.4 Sign In Page	5
2.1.5 User Authentication	6
2.1.6 User Groups	10
2.1.7 Log Out Page	11
2.2 Login Web Page HTML/CSS Code	11
2.2.1 Log In HTML Page	11
2.2.2 Register HTML Page	12
2.2.3 Logged Out HTML Page	13
<b>3. Sales</b>	<b>13</b>
3.1 Sales Page Screenshots	13
3.1.1 Creating a new work order form	13
3.1.2 All Work Order Forms	15
3.1.3 Specific Work Order Form	16
3.2 Sales Page Code	17
3.1.1 New Work Order Form html code	17
3.1.2 Work Order forms Homepage code	18
3.1.3 Specific Work Order Form Code	19
<b>4. Claims</b>	<b>20</b>
4.1 New Claim Entry Webpage	20
4.1.1 New Claim Tab	20
4.1.2 Claim Information	20
4.1.3 Date Selection	21
4.1.4 Status Dropdown	21
4.1.5 Status Approved	22
4.1.6 Status Rejected	22
4.1.7 Repair Shop and Customer Information	23
4.1.8 VIN Validator	23

4.1.9 Claim Problems	24
4.1.10 Problem Type	24
4.1.11 Problem Side	25
4.1.12 Additional Problems Added	25
4.1.13 Upload Claim Images	26
4.1.14 Claim Amounts	26
4.1.15 Claim Amount Entries	27
4.1.16 Tier 2 Claims	27
4.1.17 Form Validation	28
4.2 New Claim Code	29
4.2.1 newclaim.html	29
4.2.2 views.py	43
4.2.3 models.py	44
4.2.3 forms.py	46
4.3 All Claims Webpage	47
4.3.1 All Claims Tab	47
4.3.2 All Claims Default View	47
4.3.3 All Claims Expanded View	48
4.3.4 Claim Number Link	49
4.4 All Claims Code	50
4.4.1 claims.html	50
4.4.2 viewclaim.html	53
4.4.3 views.py	54
<b>5. Reports Dashboard</b>	<b>55</b>
5.1 Web Page	55
5.2 HTML Code	56
5.3 CSS Code	57
5.4 Python Code	59
5.4.1 views.py	59
5.4.2 urls.py	60

# 1. Introduction

The Build Screenshots artifact records all relevant building processes of C3's Web Portal.

## 1.1 Purpose

The main purpose of this document is to display the main functionalities of the C3 web portal. Included in this document are screenshots of the main functional areas of the web portal, which include user login and logout pages, sales and work order form pages, claim management pages, and a reporting dashboard. Accompanied with images of the web portal system are the html files used to display the information.

## 1.2 Scope

For the scope of this project, the login and user permission aspect of the code covers creation of a new account, signin and signout functionality. User authentication process will be implemented in the next phase once all code is compiled.

## 1.3 References

[https://www.youtube.com/watch?v=WuyKxdLcw3w&ab\\_channel=TechWithTim](https://www.youtube.com/watch?v=WuyKxdLcw3w&ab_channel=TechWithTim)

# 2. Login and User Permissions

## 2.1 Login Web Page

The following screenshots are the basic structure of the login portal of C3's Web Portal.

### 2.1.1 Login Portal

C3 Portal   Sales   Claims   Reports   Login   Register

Username:

Password:

## 2.1.2 Register Page

All workers can create an account using their username, first name, last name, work email address and confirming their password. There has also been a drop down option to clarify which department a user is a part of.

C3 Portal Sales Claims Reports Login Register

Username:

Password:

Confirm Password:

Email:

Department:

## 2.1.3 Account Creation

Once the account is created, you will be able to see the created account in the database.

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: auth\_user

	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1	00\$4Ef7Q6HgoOGqAZwv8spYoA\$7...	2023-04-10 07:07:51.861940	1	admin		admin@example.com	1	1	2023-03-20 17:51:10.178143	
2	00\$FzzipWwaENHAY0yVRrE5Dv\$z...	2023-04-10 07:11:44.125447	1	management		management@c3.ca	0	1	2023-04-09 20:38:05	
3	00\$gljFtWwV9Tln12BfM3ekBp\$tPq...	2023-04-10 07:10:21.771834	0	sales		sales@c3.ca	0	1	2023-04-10 06:40:53	
4	00\$8to9um8BCB0E3RUD152h0Q\$...	2023-04-10 07:11:28.273544	0	dealership		dealership@c3.ca	0	1	2023-04-10 06:42:23	

## 2.1.4 Sign In Page

Users who already have an account can login using the sign in page. A valid username and password must be inputted.

If there is an invalid information inputted. The user will be redirected to the main login page with the following error message

C3 Portal   Sales   Claims   Reports   Login   Register

Your username and password didn't match. Please try again.

- Please enter a correct username and password. Note that both fields may be case-sensitive.

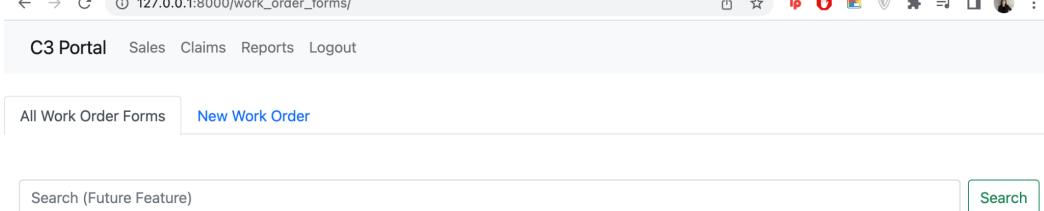
Username:

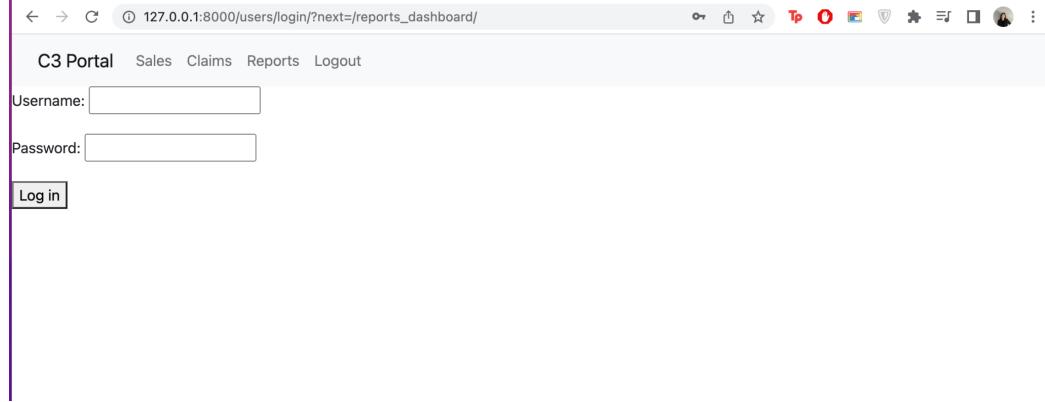
Password:

### 2.1.5 User Authentication

Once the user is logged in, they will be able to access the subsystems that they were granted access for. If they are not granted permission, they will be prompted to log in.

User with the id “sales” is under the sales department group, who can only access the sales subsystem.

Subsystem	Result																												
Sales	 <p>All Work Order Forms   <a href="#">New Work Order</a></p> <p>Search (Future Feature) <input type="button" value="Search"/></p> <table border="1"> <thead> <tr> <th>Work Order Number</th> <th>Dealership</th> <th>Date Added</th> <th>Customer</th> </tr> </thead> <tbody> <tr> <td><a href="#">Work Order Form 1</a></td> <td>Honda</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 2</a></td> <td>Toyota Dealers</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 3</a></td> <td>Audis</td> <td>April 7, 2023</td> <td>Mark</td> </tr> <tr> <td><a href="#">Work Order Form 4</a></td> <td>Toyota Dealers</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 5</a></td> <td>Honda</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 6</a></td> <td>Lindsay Kia</td> <td>April 7, 2023</td> <td>Bethany</td> </tr> </tbody> </table>	Work Order Number	Dealership	Date Added	Customer	<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark	<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany
Work Order Number	Dealership	Date Added	Customer																										
<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary																										
<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary																										
<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark																										
<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary																										
<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary																										
<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany																										

Claims	 A screenshot of a web browser showing the C3 Portal login page for the 'Reports' subsystem. The URL is 127.0.0.1:8000/users/login/?next=/reports_dashboard/. The page layout is identical to the 'Claims' page, featuring a header with 'C3 Portal' and navigation links for 'Sales', 'Claims', 'Reports', and 'Logout'. It includes 'Username:' and 'Password:' input fields and a 'Log in' button.
Reports	

User with the id “dealership” is under the dealership department group, who can only access the dealership and sales subsystem.

Subsystem	Result

Sales	<p>The screenshot shows a web browser window with the URL 127.0.0.1:8000/work_order_forms/. The page title is "C3 Portal". Below it, there are two tabs: "All Work Order Forms" (selected) and "New Work Order". A search bar labeled "Search (Future Feature)" is present. The main content is a table with columns: "Work Order Number", "Dealership", "Date Added", and "Customer". The data rows are:</p> <table border="1"> <thead> <tr> <th>Work Order Number</th><th>Dealership</th><th>Date Added</th><th>Customer</th></tr> </thead> <tbody> <tr> <td><a href="#">Work Order Form 1</a></td><td>Honda</td><td>April 7, 2023</td><td>Gary</td></tr> <tr> <td><a href="#">Work Order Form 2</a></td><td>Toyota Dealers</td><td>April 7, 2023</td><td>Gary</td></tr> <tr> <td><a href="#">Work Order Form 3</a></td><td>Audis</td><td>April 7, 2023</td><td>Mark</td></tr> <tr> <td><a href="#">Work Order Form 4</a></td><td>Toyota Dealers</td><td>April 7, 2023</td><td>Gary</td></tr> <tr> <td><a href="#">Work Order Form 5</a></td><td>Honda</td><td>April 7, 2023</td><td>Gary</td></tr> <tr> <td><a href="#">Work Order Form 6</a></td><td>Lindsay Kia</td><td>April 7, 2023</td><td>Bethany</td></tr> </tbody> </table>	Work Order Number	Dealership	Date Added	Customer	<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark	<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany																																																		
Work Order Number	Dealership	Date Added	Customer																																																																												
<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary																																																																												
<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary																																																																												
<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark																																																																												
<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary																																																																												
<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary																																																																												
<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany																																																																												
Claims	<p>The screenshot shows a web browser window with the URL 127.0.0.1:8000/claims/. The page title is "C3 Portal". Below it, there are four tabs: "All Claims" (selected), "New Claim", "Policies", and "Warranties". A search bar labeled "Search (Future Feature)" is present. The main content is a table with columns: "Claim #", "Date Added", "Status", "Claim Date", "Effective Date", "Process Date", "PO #", "File #", "Agreement #", "Shop Name", "Customer First Name", "Customer Last Name", and "P". The data rows are:</p> <table border="1"> <thead> <tr> <th>Claim #</th><th>Date Added</th><th>Status</th><th>Claim Date</th><th>Effective Date</th><th>Process Date</th><th>PO #</th><th>File #</th><th>Agreement #</th><th>Shop Name</th><th>Customer First Name</th><th>Customer Last Name</th><th>P</th></tr> </thead> <tbody> <tr> <td><a href="#">1</a></td><td>March 25, 2023</td><td>Information Only</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>456</td><td>567</td><td>678</td><td>Maple Nissan</td><td>Harry</td><td>Kalaki</td><td>In</td></tr> <tr> <td><a href="#">2</a></td><td>March 25, 2023</td><td>Rejected</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>144</td><td>87</td><td>213</td><td>Mississauga Hyundai</td><td>Johnny</td><td>Thompson</td><td>In</td></tr> <tr> <td><a href="#">3</a></td><td>March 25, 2023</td><td>Approved</td><td>March 20, 2023</td><td>March 22, 2023</td><td>March 23, 2023</td><td>3564</td><td>634</td><td>2342</td><td>Lindsay Kia</td><td>Bethany</td><td>Peters</td><td>Ex</td></tr> <tr> <td><a href="#">4</a></td><td>March 25, 2023</td><td>Pending</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>654</td><td>321</td><td>789</td><td>Alta Infiniti</td><td>Jessica</td><td>Lee</td><td>Ex</td></tr> <tr> <td><a href="#">5</a></td><td>March 25, 2023</td><td>Submitted</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>34523</td><td>2354</td><td>452</td><td>ToyotaTown</td><td>Barry</td><td>Louis</td><td>In</td></tr> </tbody> </table>	Claim #	Date Added	Status	Claim Date	Effective Date	Process Date	PO #	File #	Agreement #	Shop Name	Customer First Name	Customer Last Name	P	<a href="#">1</a>	March 25, 2023	Information Only	March 24, 2023	March 24, 2023	March 24, 2023	456	567	678	Maple Nissan	Harry	Kalaki	In	<a href="#">2</a>	March 25, 2023	Rejected	March 24, 2023	March 24, 2023	March 24, 2023	144	87	213	Mississauga Hyundai	Johnny	Thompson	In	<a href="#">3</a>	March 25, 2023	Approved	March 20, 2023	March 22, 2023	March 23, 2023	3564	634	2342	Lindsay Kia	Bethany	Peters	Ex	<a href="#">4</a>	March 25, 2023	Pending	March 24, 2023	March 24, 2023	March 24, 2023	654	321	789	Alta Infiniti	Jessica	Lee	Ex	<a href="#">5</a>	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	34523	2354	452	ToyotaTown	Barry	Louis	In
Claim #	Date Added	Status	Claim Date	Effective Date	Process Date	PO #	File #	Agreement #	Shop Name	Customer First Name	Customer Last Name	P																																																																			
<a href="#">1</a>	March 25, 2023	Information Only	March 24, 2023	March 24, 2023	March 24, 2023	456	567	678	Maple Nissan	Harry	Kalaki	In																																																																			
<a href="#">2</a>	March 25, 2023	Rejected	March 24, 2023	March 24, 2023	March 24, 2023	144	87	213	Mississauga Hyundai	Johnny	Thompson	In																																																																			
<a href="#">3</a>	March 25, 2023	Approved	March 20, 2023	March 22, 2023	March 23, 2023	3564	634	2342	Lindsay Kia	Bethany	Peters	Ex																																																																			
<a href="#">4</a>	March 25, 2023	Pending	March 24, 2023	March 24, 2023	March 24, 2023	654	321	789	Alta Infiniti	Jessica	Lee	Ex																																																																			
<a href="#">5</a>	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	34523	2354	452	ToyotaTown	Barry	Louis	In																																																																			

Reports	
---------	--

User with the id “management” is under the management department group, who can access all three of the subsystems. They will be classified as the superadmin.

Subsystem	Result																												
Sales	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 25%;">Work Order Number</th> <th style="text-align: left; width: 25%;">Dealership</th> <th style="text-align: left; width: 25%;">Date Added</th> <th style="text-align: left; width: 25%;">Customer</th> </tr> </thead> <tbody> <tr> <td><a href="#">Work Order Form 1</a></td> <td>Honda</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 2</a></td> <td>Toyota Dealers</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 3</a></td> <td>Audis</td> <td>April 7, 2023</td> <td>Mark</td> </tr> <tr> <td><a href="#">Work Order Form 4</a></td> <td>Toyota Dealers</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 5</a></td> <td>Honda</td> <td>April 7, 2023</td> <td>Gary</td> </tr> <tr> <td><a href="#">Work Order Form 6</a></td> <td>Lindsay Kia</td> <td>April 7, 2023</td> <td>Bethany</td> </tr> </tbody> </table>	Work Order Number	Dealership	Date Added	Customer	<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark	<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary	<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary	<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany
Work Order Number	Dealership	Date Added	Customer																										
<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary																										
<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary																										
<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark																										
<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary																										
<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary																										
<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany																										

Claims	<p>The screenshot shows a table of claims with the following data:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Claim #</th> <th>Date Added</th> <th>Status</th> <th>Claim Date</th> <th>Effective Date</th> <th>Process Date</th> <th>PO #</th> <th>File #</th> <th>Agreement #</th> <th>Shop Name</th> <th>Customer First Name</th> <th>Customer Last Name</th> <th>Type</th> </tr> </thead> <tbody> <tr><td>1</td><td>March 25, 2023</td><td>Information Only</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>456</td><td>567</td><td>678</td><td>Maple Nissan</td><td>Harry</td><td>Kalaki</td><td>Int</td></tr> <tr><td>2</td><td>March 25, 2023</td><td>Rejected</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>144</td><td>87</td><td>213</td><td>Mississauga Hyundai</td><td>Johnny</td><td>Thompson</td><td>Int</td></tr> <tr><td>3</td><td>March 25, 2023</td><td>Approved</td><td>March 20, 2023</td><td>March 22, 2023</td><td>March 23, 2023</td><td>3564</td><td>634</td><td>2342</td><td>Lindsay Kia</td><td>Bethany</td><td>Peters</td><td>Ex</td></tr> <tr><td>4</td><td>March 25, 2023</td><td>Pending</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>654</td><td>321</td><td>789</td><td>Alta Infiniti</td><td>Jessica</td><td>Lee</td><td>Ex</td></tr> <tr><td>5</td><td>March 25, 2023</td><td>Submitted</td><td>March 24, 2023</td><td>March 24, 2023</td><td>March 24, 2023</td><td>34523</td><td>2354</td><td>452</td><td>ToyotaTown</td><td>Barry</td><td>Louis</td><td>Int</td></tr> </tbody> </table>	Claim #	Date Added	Status	Claim Date	Effective Date	Process Date	PO #	File #	Agreement #	Shop Name	Customer First Name	Customer Last Name	Type	1	March 25, 2023	Information Only	March 24, 2023	March 24, 2023	March 24, 2023	456	567	678	Maple Nissan	Harry	Kalaki	Int	2	March 25, 2023	Rejected	March 24, 2023	March 24, 2023	March 24, 2023	144	87	213	Mississauga Hyundai	Johnny	Thompson	Int	3	March 25, 2023	Approved	March 20, 2023	March 22, 2023	March 23, 2023	3564	634	2342	Lindsay Kia	Bethany	Peters	Ex	4	March 25, 2023	Pending	March 24, 2023	March 24, 2023	March 24, 2023	654	321	789	Alta Infiniti	Jessica	Lee	Ex	5	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	34523	2354	452	ToyotaTown	Barry	Louis	Int
Claim #	Date Added	Status	Claim Date	Effective Date	Process Date	PO #	File #	Agreement #	Shop Name	Customer First Name	Customer Last Name	Type																																																																			
1	March 25, 2023	Information Only	March 24, 2023	March 24, 2023	March 24, 2023	456	567	678	Maple Nissan	Harry	Kalaki	Int																																																																			
2	March 25, 2023	Rejected	March 24, 2023	March 24, 2023	March 24, 2023	144	87	213	Mississauga Hyundai	Johnny	Thompson	Int																																																																			
3	March 25, 2023	Approved	March 20, 2023	March 22, 2023	March 23, 2023	3564	634	2342	Lindsay Kia	Bethany	Peters	Ex																																																																			
4	March 25, 2023	Pending	March 24, 2023	March 24, 2023	March 24, 2023	654	321	789	Alta Infiniti	Jessica	Lee	Ex																																																																			
5	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	34523	2354	452	ToyotaTown	Barry	Louis	Int																																																																			
Reports	<p>The dashboard has two main sections:</p> <ul style="list-style-type: none"> <li><b>Sales:</b> A table with columns for Dealership, Daily Sales, and Total Revenue.</li> <li><b>Claims:</b> A section titled "Number of Claims" with a text input field.</li> </ul>																																																																														

## 2.1.6 User Groups

User groups are managed by superadmin accounts, which would be the accounts that are within the management department. For the scope of this project, the groups will be split into two - dealership and sales.

The screenshot shows the Django administration interface for the 'Groups' model. The left sidebar has sections for 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'WEB\_APP' (Claims). The main area title is 'Select group to change'. It includes a search bar, an action dropdown set to 'GROUP', and two selected items: 'dealership' and 'sales'. A note at the bottom says '2 groups'.

## 2.1.7 Log Out Page

When user presses log out, the following message will be displayed.

The screenshot shows a simple web page with a header 'C3 Portal' and navigation links for Sales, Claims, Reports, Login, and Register. The main content area displays the message 'You have been logged out. Thank you for visiting!'

## 2.2 Login Web Page HTML/CSS Code

The following screenshots are the basic structure of the login portal of C3's Web Portal.

### 2.2.1 Log In HTML Page

```
users > templates > registration > login.html > ...
1  {% extends "web_app/base.html" %}
2
3  {% block content %}
4      {% if form.errors %}
5          <p>Your username and password didn't match. Please try again.</p>
6      {% endif %}
7
8      <form method="post" action="{% url 'users:login' %}">
9          {% csrf_token %}
10         {{ form.as_p }}
11         <button name="submit">Log in</button>
12         <input type="hidden" name="next"
13             value="{{next}} />
14     </form>
15  {% endblock content %}
16  |
```

## 2.2.2 Register HTML Page

```
users > templates > registration > register.html > ...
1  {% extends "web_app/base.html" %}
2
3  {% block content %}
4      <form method="post" action="{% url 'users:register' %}">
5          {% csrf_token %}
6          {{ form.as_p }}
7
8          <button name="submit">Register</button>
9          <input type="hidden" name="next" value="{% url 'web_app:index' %}" />
10     </form>
11
12  {% endblock content %}
13
```

### 2.2.3 Logged Out HTML Page

```
users > templates > registration > logged_out.html > ...
1  {% extends "web_app/base.html" %}
2
3  {% block content %}
4      <p>You have been logged out. Thank you for visiting!</p>
5  {% endblock content %}
6
```

## 3. Sales

The following images are screenshots from the C3 web portals sales system. The system focuses on the work order forms used by both the dealerships and C3 to transfer sales information such as C3 products that a customer purchased alongside a vehicle from the dealership.

### 3.1 Sales Page Screenshots

#### 3.1.1 Creating a new work order form

These images display the work order form page that dealerships will use to create a new work order form and submit into the C3 portal.

C3 Portal Work Order Forms

All Work Order Forms New Work Order Form

## New Work Order Form

Enter a new Work Order Form by filling out the information below.

---

### Work Order Form Information

Work Order Form #	Date Added	
<input type="text"/>	March 29, 2023	
Product Used	Quantity Used	Payment Amount
<input type="text"/> Choose...	<input type="text"/>	<input type="text"/>

---

### Customer Information

Customer Name	Phone Number
<input type="text"/>	<input type="text"/>
Vehicle Make	Vehicle VIN
<input type="text"/>	<input type="text"/>

---

### Dealership Information

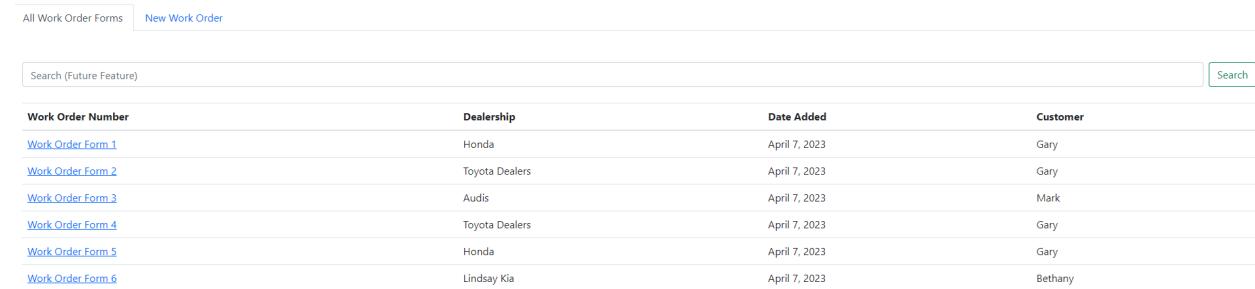
Dealership Name	Phone Number	
<input type="text"/>	<input type="text"/> 123-456-7890	
Contact Name (Optional)	Email (Optional)	
<input type="text"/>	<input type="text"/> you@example.com	
Address		
<input type="text"/> 1234 Main St		
Address 2 (Optional)		
<input type="text"/> Apartment or Suite		
Country	Province/Territory	Postal Code
<input type="text"/> Canada	<input type="text"/> Choose...	<input type="text"/> A1A 1A1

---

**Submit**

### 3.1.2 All Work Order Forms

These screenshots display the homepage that C3 management will use to view all work order forms. Each work order form is hyperlinked so users can click on a specific form to see the exact details of each form.



The screenshot shows a web-based application interface for managing work orders. At the top, there are two buttons: "All Work Order Forms" (highlighted in blue) and "New Work Order". Below the buttons is a search bar labeled "Search (Future Feature)" and a "Search" button. The main content area displays a table with four columns: "Work Order Number", "Dealership", "Date Added", and "Customer". The table contains six rows of data, each representing a work order entry. The data is as follows:

Work Order Number	Dealership	Date Added	Customer
<a href="#">Work Order Form 1</a>	Honda	April 7, 2023	Gary
<a href="#">Work Order Form 2</a>	Toyota Dealers	April 7, 2023	Gary
<a href="#">Work Order Form 3</a>	Audis	April 7, 2023	Mark
<a href="#">Work Order Form 4</a>	Toyota Dealers	April 7, 2023	Gary
<a href="#">Work Order Form 5</a>	Honda	April 7, 2023	Gary
<a href="#">Work Order Form 6</a>	Lindsay Kia	April 7, 2023	Bethany

### 3.1.3 Specific Work Order Form

This is a screenshot of a specific work order form that will contain all the necessary information for C3's sales system to function correctly.

All Work Orders    Current Work Order Form    [New Work Order Form](#)

## Work Order Form 1

### Work Order Form Information

---

**Date Added:**  
April 7, 2023, 4:33 a.m.

**Product:**  
status\_info

**Quantity Used:**  
1

**Payment Amount:**  
200

### Customer Information

---

**Customer Name:**  
Gary

**Customer Phone:**  
7802224242

**Vehicle Make:**  
Honda

**Vehicle VIN:**  
12345

### Dealership Information

---

**Dealership Name:**  
Honda

**Dealership Phone:**  
7802224444

**Contact Name:**  
Gerald

**Dealership Email:**  
gary@hondadealers.com

**Address:**  
43212 Main Street

**Country:**  
CAN

**Province:**  
AB

**Postal Code:**  
T6V 1R1

## 3.2 Sales Page Code

### 3.1.1 New Work Order Form html code

```

<% extends "web_app/base.html" %>
<% block content %>


<nav class="nav nav-tabs">
  <li class="nav-item" role="presentation">
    <a class="nav-link" href="{% url 'web_app:work_order_forms' %}">All Work Order Forms</a>
  </li>
  <li class="nav-item" role="presentation">
    <button class="nav-link active" id="allworkorders-tab" data-bs-toggle="tab" href="{% url 'web_app:new_work_order_form' %}" role="tab" aria-controls="nav-newclaim" aria-selected="true">New Work Order Form</button>
  </li>
</ul> <!-- END - Navigation Tabs -->

<br>


<div class="container">

  <!-- Heading -->
  <div class="py-5 text-center">
    <h2>New Work Order Form</h2>
    <p class="lead">Enter a new Work Order Form by filling out the information below.</p>
  </div>

  <!-- Form -->
  <div class="col-md-15 order-md-0">
    <form method="post" action="{% url 'web_app:new_work_order_form' %}" id="Work_order_form_form" class="needs-validation" novalidate>
      {% csrf_token %}

      <hr class="mb-4"> <!-- Divider Line -->
      <!-- WOF Information Row -->
      <h4 class="mb-3 text-center">Work Order Form Information</h4>
      <div class="row justify-content-center">

        <!-- Work Order Form # -->
        <div class="col-md-3 mb-3">
          <label for="work_order_form_number">Work Order Form #</label>

```

### 3.1.2 Work Order forms Homepage code

```

<% extenus web_app/base.html %>

{% block content %}
<br>
<!-- Navigation Tabs, source: https://getbootstrap.com/docs/5.0/components/navs-tabs/ -->


- All Work Order Forms</button>
- New Work Order</a>


</ul>
<br>

<div style="margin-left:16px; margin-right:16px">
<br>
<form class="d-flex" role="search">
<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
<button class="btn btn-outline-success" type="submit">Search</button>
</form>
<br>

<button id="toggle-button">Show/Hide Additional Work Order Form Information</button>
<br>
<br>

<!-- Work Order Forms Table -->
<table class="table">
<thead>
<tr>
<th>Work Order Number</th>
<th>Dealership</th>
<th>Date Added</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="{% url 'web_app:work_order_form' work_order_form.id %}">Work Order Form {{ work_order_form.id }}</a></td> <!--creates link to specific WOF-->
<td>{{ work_order_form.dealership }}</td>
<td>{{ work_order_form.date_added|date:"F j, Y" }}</td>
<td>{{ work_order_form.status }}</td>
</tr>
<% endif %>
</tbody>
</table>
</div>
{% endblock content %}

```

### 3.1.3 Specific Work Order Form Code

```

<% calendar_new_app_base.html %>
{% block content %}

<!-- Navigation Tabs, source: https://getbootstrap.com/docs/4.0/components/navbar/ --&gt;
<ul class="nav nav-tabs">
- All Work Orders
- Current Work Order Form
- New Work Order Form


<!-- END - Navigation Tabs --&gt;
&lt;br&gt;

<!-- Work Order Form table --&gt;
&lt;h1&gt;Work Order Form: {{ work_order_form.id }}&lt;/h1&gt;

&lt;hr class="mb-4"&gt; &lt!-- Divider Line --&gt;

&lt;h2&gt;Work Order Information:&lt;/h2&gt;
&lt;table class="table table-striped"&gt;
&lt;thead&gt;
&lt;tr&gt;
&lt;th scope="col"&gt;Date Added&lt;/th&gt;
&lt;th scope="col"&gt;Product Used&lt;/th&gt;
&lt;th scope="col"&gt;Quantity Used&lt;/th&gt;
&lt;th scope="col"&gt;Payment Amount&lt;/th&gt;
&lt;th scope="col"&gt;Title&lt;/th&gt;
&lt;th scope="col"&gt;Description&lt;/th&gt;
&lt;!-- Add more columns as needed --&gt;
&lt;/tr&gt;
&lt;/thead&gt;
&lt;tbody&gt;
{% for entry in entries %}
</pre>

```

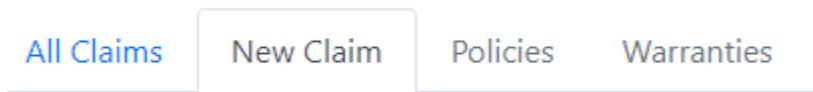
## 4. Claims

The following screenshots are the structure of the claims in C3's Web Portal.

### 4.1 New Claim Entry Webpage

#### 4.1.1 New Claim Tab

Navigate to the New Claim tab to enter a new claim into the system.



#### 4.1.2 Claim Information

General claim information can be found at the top of the new claim entry. The claim number and date added are automatically generated for each new claim. Other entries include the claim status, claim date, effective date, process date, PO number, file number, and agreement number.

### New Claim

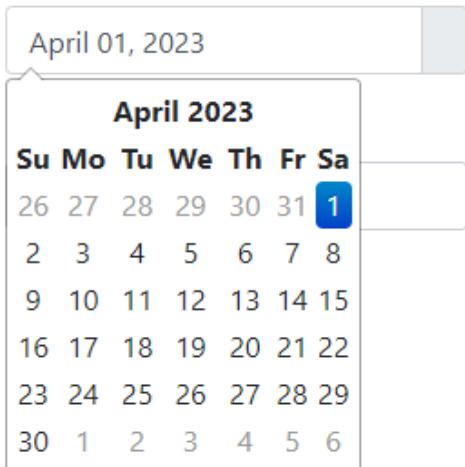
Enter a new claim by filling out the information below.

Claim #	Date Added	Status
9	March 25, 2023	Choose...
Claim Date	Effective Date	Process Date
March 24, 2023	March 24, 2023	March 24, 2023
PO #	File #	Agreement #

#### 4.1.3 Date Selection

For claim date, effective date, and process date, the date is automatically filled in as the current date. The user can change the date by choosing a date on the calendar popup that appears when clicking on the date.

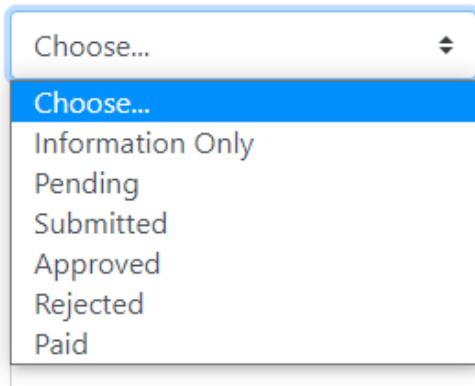
Claim Date



#### 4.1.4 Status Dropdown

In the status dropdown menu, the user can select the status of the claim.

Status



#### 4.1.5 Status Approved

If ‘Approved’ was selected for the claim status, the approval entries are shown for the user to fill out. This section is hidden by default.

### New Claim

Enter a new claim by filling out the information below.

Claim #	Date Added	Status
9	March 25, 2023	Approved
Claim Date	Effective Date	Process Date
March 24, 2023	March 24, 2023	March 24, 2023
PO #	File #	Agreement #

### Approval

Approved By	Approved To	Approved Date
		March 24, 2023

#### 4.1.6 Status Rejected

If ‘Rejected’ was selected for the claim status, the rejection entries are shown for the user to fill out. This section is hidden by default.

### New Claim

Enter a new claim by filling out the information below.

Claim #	Date Added	Status
9	March 25, 2023	Rejected
Claim Date	Effective Date	Process Date
March 24, 2023	March 24, 2023	March 24, 2023
PO #	File #	Agreement #

### Rejected

Rejected By	Rejection Date
	March 24, 2023

#### 4.1.7 Repair Shop and Customer Information

Following the general claim information section are the repair shop and customer information sections. The repair shop phone number and postal code automatically formats the user's entry to the correct formats.

##### Repair Shop Information

<input type="checkbox"/> Active			
Shop Name	Phone Number		
<input type="text"/>	<input type="text" value="123-456-7890"/>		
Contact Name (Optional)	Email (Optional)		
<input type="text"/>	<input type="text" value="you@example.com"/>		
Address			
<input type="text" value="1234 Main St"/>			
Address 2 (Optional)			
<input type="text" value="Apartment or Suite"/>			
Country	Province/Territory	Postal Code	
<input type="text" value="Canada"/>	<input type="text" value="Choose..."/>	<input type="text" value="A1A 1A1"/>	

##### Customer Information

First Name	Last Name	Policy Type	Coverage Type
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Vehicle VIN	Make (Future Feature)	Model (Future Feature)	Year (Future Feature)
<input type="text"/>			

#### 4.1.8 VIN Validator

When a user enters the vehicle's vehicle information number (VIN), it is validated to ensure a valid VIN is entered. A feature that can be implemented in the future, is automatically filling in the vehicle's make, model, and year from the VIN entered.

##### Customer Information

First Name	Last Name	Policy Type	Coverage Type
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Vehicle VIN	Make (Future Feature)	Model (Future Feature)	Year (Future Feature)
<input style="border: 2px solid red; padding: 2px; width: 50px; height: 20px; text-align: center; vertical-align: middle; font-size: 1em; margin-right: 10px;" type="text" value="123"/>			
Valid VIN required.			

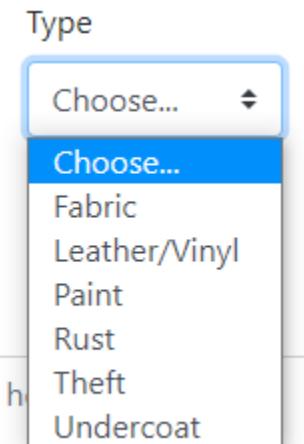
#### 4.1.9 Claim Problems

Following the customer information section, is the section to fill out the problem(s) for the claim. This section includes entries for the problem(s), a textbox to enter in a detailed explanation (textbox size can be expanded from the bottom-right corner), and a section to upload claim images to the entry. Saving the problem type, description, part, side, and goodwill to the database is in development and not implemented in this prototype.

Problem				
Type	Description	Part	Side	Goodwill
<input type="button" value="Choose..."/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Choose..."/>	\$ <input type="text" value="0.00"/> <span>-</span> <span>+</span>
<b>Details</b> <input type="text" value="Enter text here"/>				
<b>Upload Claim Images</b> <input type="button" value="Choose Files"/> No file chosen				

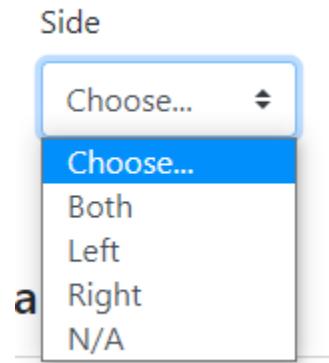
#### 4.1.10 Problem Type

The user can select the problem type from the Type dropdown menu.



#### 4.1.11 Problem Side

The user can select the side of where the problem occurs from the Side dropdown menu.



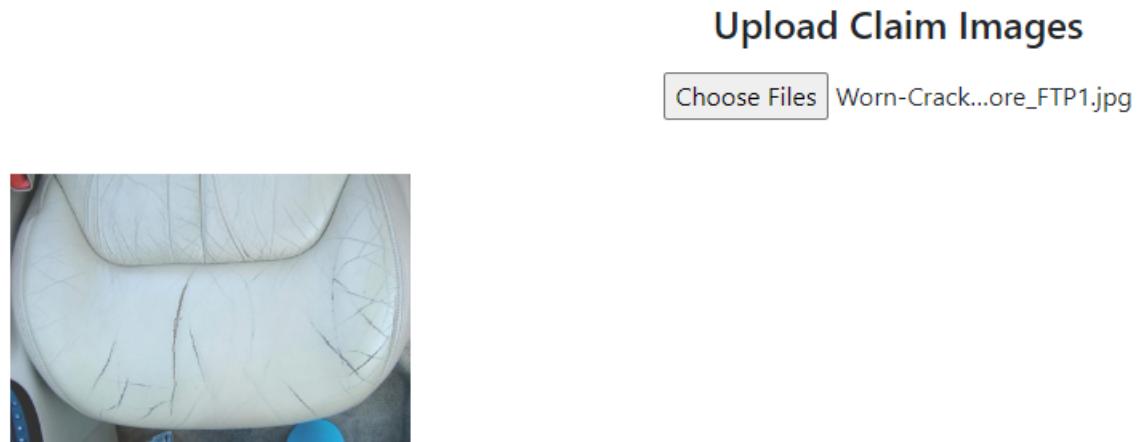
#### 4.1.12 Additional Problems Added

By default, only one problem entry is shown on the new claim form. The user can add additional problems to the claim by pressing the '+' button. The new problem entry row appears below whichever row the '+' button was pressed. To remove a problem, the user can press the '-' button on whichever row they want removed. Each claim must have at least one problem, so the user is restricted from removing the last remaining row.

Problem				
Type	Description	Part	Side	Goodwill
<input type="button" value="Choose..."/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Choose..."/>	\$ 0.00
<input type="button" value="-"/>	<input type="button" value="+"/>			
<input type="button" value="Choose..."/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Choose..."/>	\$ 0.00
<input type="button" value="-"/>	<input type="button" value="+"/>			
<input type="button" value="Choose..."/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Choose..."/>	\$ 0.00
<input type="button" value="-"/>	<input type="button" value="+"/>			

#### 4.1.13 Upload Claim Images

Users can upload images to the new claim entry by clicking on the Choose Files button and selecting images from their device. Multiple images can be selected at the same time. Saving the uploaded images in a file for the claim is currently not implemented in this prototype.



#### 4.1.14 Claim Amounts

Following the problem section is the claim amounts section. This section includes the claim estimates, actual claim amounts, and payment information. The actual claim net amount is automatically calculated. This is the last section for the user to fill out so the submit button is shown right after.

<b>Estimates</b>			
<input type="checkbox"/> Claim Over \$1500	Estimate #1	Estimate #2	
	\$ 0.00	\$ 0.00	
		Requested	
		\$ 0.00	
<b>Actual Claim Amount</b>			
Agreed Amount	Deductible	Net Amount	
\$ 0.00	\$ 0.00	\$ 0.00	
<b>Payment Information</b>			
Cheque #	Pay To	Cheque Date	Cheque Amount
<input type="text"/>	<input type="text"/>	<input type="text" value="March 24, 2023"/>	<input type="text" value="\$ 0.00"/>
<b>Submit</b>			

#### 4.1.15 Claim Amount Entries

When entering a dollar amount into any of the claim amount entries, the user can type in a number or adjust the amount via the arrows in the entry box. The user's entry is automatically formatted and rounded to the nearest cent.

Estimate #1

\$	0.00	▲ ▼
----	------	-----

#### 4.1.16 Tier 2 Claims

When a claim is over \$1500, the claim becomes a tier 2 claim and additional estimates are required. Additional estimates are hidden by default and can be shown by selecting the the 'Claim Over \$1500' checkbox.

#### Estimates

Claim Over \$1500

Estimate #1

\$	0.00
----	------

Estimate #2

\$	0.00
----	------

Requested

\$	0.00
----	------

Estimate #3

\$	0.00
----	------

Estimate #4

\$	0.00
----	------

Estimate #5

\$	0.00
----	------

#### 4.1.17 Form Validation

When the submit button is pressed, the form is validated and will only be entered into the system when all required fields are filled in. The user can easily see which fields are missing through the additional formatting that appears when submitting an incomplete form. In this example, the status ‘Rejected’ was chosen which makes the rejected section required for the user to complete the entries. This section is not required when another status is selected. The same concept applies for when the user selects ‘Approved’ and the approval section appears.

## New Claim

Enter a new claim by filling out the information below.

Claim #	Date Added	Status
9	March 25, 2023	Rejected ✓
Claim Date	Effective Date	Process Date
March 24, 2023 ✓	March 24, 2023 ✓	March 24, 2023 ✓
PO #	File #	Agreement #
<span style="border: 1px solid red; padding: 2px;"> </span> ✗	<span style="border: 1px solid red; padding: 2px;"> </span> ✗	<span style="border: 1px solid red; padding: 2px;"> </span> ✗
PO # required.	File # required.	Agreement required.

## Rejected

Rejected By	Rejection Date
<span style="border: 1px solid red; padding: 2px;"> </span> ✗	March 24, 2023 ✓
Name required.	
Reason for Rejection	
<span style="border: 1px solid red; padding: 10px;">Enter text here ✗</span>	
Reason required.	

## 4.2 New Claim Code

### 4.2.1 newclaim.html

```

1  <!-- Anything without a source specified was learned from changes and https://getbootstrap.com/docs/4.0/examples -->
2
3  {% extends "web_app/base.html" %} 
4
5  {% block content %}
6
7
8  <br>
9
10 <!-- Navigation Tabs source: https://getbootstrap.com/docs/4.0/components/navbar/ -->
11 <ul class="nav nav-tabs" role="presentation">
12   <li class="nav-item" role="presentation">
13     <a class="nav-link" href="{% url 'web_app:claims' %}">All Claims</a>
14   </li>
15   <li class="nav-item" role="presentation">
16     <a href="#" class="nav-link active" id="allclaims-tab" data-bs-toggle="tab" href="{% url 'web_app:newclaim' %}" role="tab" aria-controls="nav-newclaim" aria-selected="true">New Claim</a>
17   </li>
18   <li class="nav-item" role="presentation">
19     <a class="nav-link disabled" href="#">Policies</a>
20   </li>
21   <li class="nav-item" role="presentation">
22     <a class="nav-link disabled" href="#">Warranties</a>
23   </li>
24 </ul> <!-- END - Navigation Tabs -->
25
26 <br>
27
28 <!-- New Claim Form, source: https://getbootstrap.com/docs/4.0/examples-->
29 <div class="container">
30   <!-- Heading -->
31   <div class="py-5 text-center">
32     <h2>New Claim</h2>
33     <p><span>lead</span> Enter a new claim by filling out the information below.</p>
34   </div>
35
36   <!-- Form -->
37   <div class="col-md-15 order-md-0">
38     <form method="post" action="{% url 'web_app:newclaim' %}" id="claimForm" class="needs-validation" novalidate>
39       <input type="hidden" name="csrf_token" value="{{ csrf_token }}"/>
40
41       <!-- Claim Information Row -->
42       <div class="row justify-content-center">
43
44         <!-- Claim # -->
45         <div class="col-md-3 mb-3">
46           <label for="claim_number">Claim #</label>
47           <input type="text" class="form-control" id="claim_number" name="claim_number" value="{{ claim_number }}" disabled>
48         </div> <!-- END - Claim # -->
49
50         <!-- Date Added -->
51         <div class="col-md-3 mb-3">
52           <label for="date_added">Date Added</label>
53           <div class="input-group">
54             <input type="text" class="form-control" id="date_added" name="date_added" value="{{ formatted_date }}" required disabled>
55             <div class="input-group-append">
56               <span class="input-group-text"><i class="fa fa-calendar"></i></span>
57             </div>
58           </div>
59         </div>
60
61         <script>
62           // Get the current date
63           var currentDate = new Date();
64
65           // Set the value of the claim date field to the formatted date
66           var options = { month: 'long', day: 'numeric', year: 'numeric' };
67           var formattedDate = currentDate.toLocaleDateString('en-US', options);
68           document.getElementById('date_added').value = formattedDate;
69
70           // Initialize the datepicker
71           $('.input-group.date').datepicker({
72             format: 'MM dd, yyyy',
73             todayHighlight: true,
74             autoClose: true,
75             orientation: 'bottom',
76             templates: {
77               leftArrow: '<i class="fa fa-angle-left"></i>',
78               rightArrow: '<i class="fa fa-angle-right"></i>'
79             }
80           });
81         </script> <!-- END - Date Added -->
82
83         <!-- Status -->
84         <div class="col-md-3 mb-3">
85           <label for="status">Status</label>
86           <select class="custom-select d-block w-100" id="status" name="status" required>
87             <option value="">Choose...</option>
88             <option value="status_info">Information Only</option>
89             <option value="status_pending">Pending</option>
90             <option value="status_submitted">Submitted</option>
91             <option value="status_approved">Approved</option>
92             <option value="status_rejected">Rejected</option>
93             <option value="status_paid">Paid</option>
94           </select>
95           <div class="invalid-feedback">
96             Please select a status.
97           </div>
98         </div> <!-- END - Status -->
99
100       </div> <!-- END - Claim Information Row -->
101
102   </div>

```

```

184      <!-- Dates Row -->
185      <div class="row justify-content-center">
186
187          <!-- Include the Bootstrap and jQuery libraries -->
188          <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/css/bootstrap-datepicker.min.css">
189          <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
190          <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
191          <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
192
193          <!-- Include the Bootstrap datepicker library -->
194          <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/js/bootstrap-datepicker.min.js"></script>
195
196          <!-- Claim Date -->
197          <div class="col-md-3 mb-3">
198              <label for="claim_date">Claim Date</label>
199              <div class="input-group">
200                  <input type="text" class="form-control" id="claim_date" name="claim_date" value="{{ formatted_date }}" required>
201                  <div class="input-group-append">
202                      <span class="input-group-text"><i class="fa fa-calendar"></i></span>
203                  </div>
204              </div>
205          </div> <!-- END - Claim Date -->
206
207          <script>
208              // Get the current date
209              var currentDate = new Date();
210
211              // Set the value of the claim date field to the formatted date
212              var options = { month: 'long', day: 'numeric', year: 'numeric' };
213              var formattedDate = currentDate.toLocaleDateString('en-US', options);
214              document.getElementById("claim_date").value = formattedDate;
215
216              // Initialize the datepicker
217              $('.input-group.date').datepicker({
218                  format: 'MM dd, yyyy',
219                  todayHighlight: true,
220                  autoclose: true,
221                  orientation: 'bottom',
222                  templates: {
223                      leftArrow: '<i class="fa fa-angle-left"></i>',
224                      rightArrow: '<i class="fa fa-angle-right"></i>'
225                  }
226              });
227
228      </script>
229
230      <!-- Effective Date -->
231      <div class="col-md-3 mb-3">
232          <label for="effective_date">Effective Date</label>
233          <div class="input-group">
234              <input type="text" class="form-control" id="effective_date" name="effective_date" value="{{ formatted_date }}" required>
235              <div class="input-group-append">
236                  <span class="input-group-text"><i class="fa fa-calendar"></i></span>
237              </div>
238          </div>
239      </div> <!-- END - Effective Date -->
240
241
242
243      <script>
244          // Get the current date
245          var currentDate = new Date();
246
247          // Set the value of the claim date field to the formatted date
248          var options = { month: 'long', day: 'numeric', year: 'numeric' };
249          var formattedDate = currentDate.toLocaleDateString('en-US', options);
250          document.getElementById("effective_date").value = formattedDate;
251
252          // Initialize the datepicker
253          $('.input-group.date').datepicker({
254              format: 'MM dd, yyyy',
255              todayHighlight: true,
256              autoclose: true,
257              orientation: 'bottom',
258              templates: {
259                  leftArrow: '<i class="fa fa-angle-left"></i>',
260                  rightArrow: '<i class="fa fa-angle-right"></i>'
261              }
262          });
263
264
265      </script>
266
267      <!-- Process Date -->
268      <div class="col-md-3 mb-3">
269          <label for="process_date">Process Date</label>
270          <div class="input-group">
271              <input type="text" class="form-control" id="process_date" name="process_date" value="{{ formatted_date }}" required>
272              <div class="input-group-append">
273                  <span class="input-group-text"><i class="fa fa-calendar"></i></span>
274              </div>
275          </div>
276      </div>
277
278      <script>
279          // Get the current date
280          var currentDate = new Date();
281
282          // Set the value of the claim date field to the formatted date
283          var options = { month: 'long', day: 'numeric', year: 'numeric' };
284          var formattedDate = currentDate.toLocaleDateString('en-US', options);
285          document.getElementById("process_date").value = formattedDate;
286
287          // Initialize the datepicker
288          $('.input-group.date').datepicker({
289              format: 'MM dd, yyyy',
290              todayHighlight: true,
291              autoclose: true,
292              orientation: 'bottom',
293              templates: {
294                  leftArrow: '<i class="fa fa-angle-left"></i>',
295                  rightArrow: '<i class="fa fa-angle-right"></i>'
296              }
297          });
298
299      </script> <!-- END - Process Date -->
300
301  </div> <!-- END - Dates Row -->

```

```

216
217     <!-- Order Information Row -->
218     <div class="row justify-content-center">
219
220         <!-- PO Number -->
221         <div class="col-md-3 mb-3">
222             <label for="po_num">PO #</label>
223             <input type="text" class="form-control" id="po_num" name="po_num" placeholder="" required>
224             <div class="invalid-feedback">
225                 PO # required.
226             </div>
227         </div> <!-- END - PO Number -->
228
229         <!-- File Number -->
230         <div class="col-md-3 mb-3">
231             <label for="file_num">File #</label>
232             <input type="text" class="form-control" id="file_num" name="file_num" placeholder="" required>
233             <div class="invalid-feedback">
234                 File # required.
235             </div>
236         </div> <!-- END - File Number -->
237
238         <!-- Inspection Agreement # -->
239         <div class="col-md-3 mb-3">
240             <label for="agreement">Agreement #</label>
241             <input type="text" class="form-control" id="agreement" name="agreement" placeholder="" required>
242             <div class="invalid-feedback">
243                 Agreement required.
244             </div>
245         </div> <!-- END - Inspection Agreement # -->
246
247
248         <br>
249         <br>
250
251         <!-- Approval Row (hidden unless approved selected in status)-->
252
253         <!-- Hide/Show Approval Row -->
254         <script>
255             $(document).ready(function() {
256                 // Hide the Approval section initially
257                 $('#approval-section').hide();
258
259                 // Add an event listener to the Status field
260                 $('#status').on('change', function() {
261                     var status = $(this).val();
262                     if (status === 'status_approved') {
263                         $('#approval-section').show();
264                     } else {
265                         $('#approval-section').hide();
266                     }
267                 });
268             });
269         </script> <!-- END - Hide/Show Approval Row -->
270
271         <div id="approval-section">
272             <h4 class="mb-3 text-center">Approval</h4>
273             <div class="row justify-content-center">
274
275                 <!-- Approved By -->
276                 <div class="col-md-3 mb-3">
277                     <label for="claim_approvedby">Approved By:</label>
278                     <input type="text" class="form-control" id="claim_approvedby" name="claim_approvedby" placeholder="" required>
279                     <div class="invalid-feedback">
280                         Name required.
281                     </div>
282                 </div> <!-- END - Approved By -->
283
284                 <!-- Approved To -->
285                 <div class="col-md-3 mb-3">
286                     <label for="claim_approvedto">Approved To:</label>
287                     <input type="text" class="form-control" id="claim_approvedto" name="claim_approvedto" placeholder="" required>
288                     <div class="invalid-feedback">
289                         Name required.
290                     </div>
291                 </div> <!-- END - Approved To -->
292
293                 <!-- Include the Bootstrap and jQuery libraries -->
294                 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/css/bootstrap-datepicker.min.css" type="text/css">
295                 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
296                 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
297                 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
298
299                 <!-- Include the Bootstrap datepicker library -->
300                 <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/js/bootstrap-datepicker.min.js"></script>
301
302                 <!-- Approved Date -->
303                 <div class="col-md-3 mb-3">
304                     <label for="claim_approved_date">Approved Date:</label>
305                     <div class="input-group date">
306                         <input type="text" class="form-control" id="claim_approved_date" name="claim_approved_date" value="{{ formatted_date }}">
307                         <div class="input-group-append">
308                             <span class="input-group-text"><i class="fa fa-calendar"></i></span>
309                         </div>
310                     </div>
311                 </div>
312
313
314
315
316
317
318

```

```

320         <script>
321             // Get the current date
322             var currentDate = new Date();
323
324             // Set the value of the claim date field to the formatted date
325             var options = { month: 'long', day: 'numeric', year: 'numeric' };
326             var formattedDate = currentDate.toLocaleDateString('en-US', options);
327             document.getElementById("claim_approved_date").value = formattedDate;
328
329             // Initialize the datepicker
330             $('.input-group.date').datepicker({
331                 format: 'MM dd, yyyy',
332                 todayHighlight: true,
333                 autoclose: true,
334                 orientation: 'bottom',
335                 templates: {
336                     leftArrow: '<i class="fa fa-angle-left"></i>',
337                     rightArrow: '<i class="fa fa-angle-right"></i>'
338                 }
339             });
340             </script> <!-- END - Approved Date -->
341         </div> <!-- END - Approval Row -->
342
343         <!-- Approval Entries Required when Approved is selected in Status -->
344         <script>
345             $('#status').on('change', function() {
346                 if ($(this).val() === 'status_approved') {
347                     $('.approval-section').show();
348                     $('#claim_approvedby').attr('required', true);
349                     $('#claim_approvedto').attr('required', true);
350                     $('#claim_approved_date').attr('required', true);
351                 } else {
352                     $('.approval-section').hide();
353                     $('#claim_approvedby').removeAttr('required');
354                     $('#claim_approvedto').removeAttr('required');
355                     $('#claim_approved_date').removeAttr('required');
356                 }
357             });
358             </script> <!-- END - Approval Entries Required when Approved is selected in Status -->
359
360             <br>
361             <br>
362         </div> <!-- END - Approval Row (hidden unless approved selected in status)-->
363
364
365         <!-- Rejected Row (hidden unless rejected selected in status)-->
366
367         <!-- Hide/Show Rejected Row -->
368         <script>
369             $(document).ready(function() {
370                 // Hide the Rejected section initially
371                 $('#rejected-section').hide();
372
373                 // Add an event listener to the Status field
374                 $('#status').on('change', function() {
375                     var status = $(this).val();
376                     if (status === 'status_rejected') {
377                         $('#rejected-section').show();
378                     } else {
379                         $('#rejected-section').hide();
380                     }
381                 });
382             });
383             </script> <!-- END - Hide/Show Rejected Row -->
384
385             div id="rejected-section">
386                 <h4 class="mb-3 text-center">Rejected</h4>
387                 <div class="row justify-content-center">
388                     <!-- Rejected By -->
389                     <div class="col-md-4 mb-3">
390                         <label for="claim_rejectedby">Rejected By</label>
391                         <input type="text" class="form-control" id="claim_rejectedby" name="claim_rejectedby" placeholder="">
392                         <div class="invalid-feedback">
393                             Name required.
394                         </div>
395                     </div>
396                     </div> <!-- END - Rejected By -->
397
398                     <!-- Rejection Date -->
399                     <div class="col-md-4 mb-3">
400                         <label for="claim_rejection_date">Rejection Date:</label>
401                         <div class="input-group date">
402                             <input type="text" class="form-control" id="claim_rejection_date" name="claim_rejection_date" value="{{ formatted_date }}">
403                             <div class="input-group-append">
404                                 <span class="input-group-text"><i class="fa fa-calendar"></i></span>
405                             </div>
406                         </div>
407                     </div>
408
409                     <script>
410                         // Get the current date
411                         var currentDate = new Date();
412
413                         // Set the value of the claim date field to the formatted date
414                         var options = { month: 'long', day: 'numeric', year: 'numeric' };
415                         var formattedDate = currentDate.toLocaleDateString('en-US', options);
416                         document.getElementById("claim_rejection_date").value = formattedDate;
417
418                         // Initialize the datepicker
419                         $('.input-group.date').datepicker({
420                             format: 'MM dd, yyyy',
421                             todayHighlight: true,
422                             autoclose: true,
423                             orientation: 'bottom',
424                             templates: {
425                                 leftArrow: '<i class="fa fa-angle-left"></i>',
426                                 rightArrow: '<i class="fa fa-angle-right"></i>'
427                             }
428                         });
429                         </script> <!-- END - Rejection Date -->
430                     </div>

```

```

431         <div class="row justify-content-center">
432             <!-- Reason For Rejection -->
433             <div class="col-md-6 mb-3">
434                 <label for="claim_rejection_reason">Reason for Rejection</label>
435                 <textarea class="form-control" id="claim_rejection_reason" name="claim_rejection_reason" placeholder="Enter text here" rows="5" required></textarea>
436                 <div class="invalid-feedback">
437                     Reason required.
438                 </div>
439             </div> <!-- END - Reason For Rejection -->
440         </div>
441
442         <!-- Rejected Entries Required when Rejected is selected in Status -->
443         <script>
444             $(status).on('change', function() {
445                 if ($(this).val() === 'status_rejected') {
446                     $('.approval-section').show();
447                     $('#claim_rejectedby').attr('required', true);
448                     $('#claim_rejection_reason').attr('required', true);
449                     $('#claim_rejection_date').attr('required', true);
450                 } else {
451                     $('.approval-section').hide();
452                     $('#claim_rejectedby').removeAttr('required');
453                     $('#claim_rejection_reason').removeAttr('required');
454                     $('#claim_rejection_date').removeAttr('required');
455                 }
456             });
457         </script> <!-- END - Rejected Entries Required when Rejected is selected in Status -->
458
459         <br>
460     </div> <!-- END - Rejected Row (hidden unless rejected selected in status)-->
461
462     <!-- Dealership Information -->
463     <h4 class="mb-3 text-center">Repair Shop Information</h4>
464
465     <!-- Active -->
466     <input type="checkbox" id="shop_active" name="shop_active">
467     <label for="shop_active">Active</label>
468
469     <!-- Dealership Information Row 1 -->
470     <div class="row justify-content-center">
471         <!-- Shop Name -->
472         <div class="col-md-6 mb-3">
473             <label for="shop_name">Shop Name</label>
474             <input type="text" class="form-control" id="shop_name" name="shop_name" placeholder="" required>
475             <div class="invalid-feedback">
476                 Shop name required.
477             </div>
478         </div> <!-- END - Shop Name -->
479
480         <!-- Phone Number -->
481         <div class="col-md-6 mb-3">
482             <label for="shop_phone">Phone Number</label>
483             <input type="tel" class="form-control" id="shop_phone" name="shop_phone" pattern="[\d]{3}-[\d]{3}-[\d]{4}" placeholder="123-456-7890" required>
484             <div class="invalid-feedback">
485                 Phone number required.
486             </div>
487         </div>
488
489
490         <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
491         <script>
492             $(document).ready(function() {
493                 $('#shop_phone').on('input', function() {
494                     var input = $(this).val();
495                     if (input.match(/\D/g)) {
496                         // Remove all non-numeric characters
497                         input = input.replace(/\D/g, '');
498                         $(this).val(input);
499                     }
500                     // Format the phone number
501                     if (input.length >= 3) {
502                         input = input.slice(0, 3) + '-' + input.slice(3);
503                     }
504                     if (input.length >= 7) {
505                         input = input.slice(0, 7) + '-' + input.slice(7);
506                     }
507                     $(this).val(input);
508                 });
509             });
510         </script> <!-- END - Phone Number -->
511
512     </div> <!-- END - Dealership Information Row 1 -->
513
514
515     <!-- Dealership Information Row 2 -->
516     <div class="row justify-content-center">
517
518         <!-- Contact Name -->
519         <div class="col-md-6 mb-3">
520             <label for="shop_contact">Contact Name <span class="text-muted">(Optional)</span></label>
521             <input type="text" class="form-control" id="shop_contact" name="shop_contact" placeholder="">
522         </div> <!-- END - Contact Name -->
523
524         <!-- Email -->
525         <div class="col-md-6 mb-3">
526             <label for="shop_email">Email <span class="text-muted">(Optional)</span></label>
527             <input type="email" class="form-control" id="shop_email" name="shop_email" placeholder="you@example.com">
528             <div class="invalid-feedback">
529                 Please enter a valid email address.
530             </div>
531         </div> <!-- END - Email -->
532     </div> <!-- END - Dealership Information Row 2 -->
533
534
535     <!-- Dealership Information Row 3- -->
536     <!-- Address -->
537     <div class="mb-2">
538         <label for="shop_address">Address</label>
539         <input type="text" class="form-control" id="shop_address" name="shop_address" placeholder="1234 Main St" required>
540         <div class="invalid-feedback">
541             Please enter your shipping address.
542         </div>
543     </div> <!-- END - Address -->

```

```

545      <!-- Dealership Information Row 4-->
546      <!-- Address 2 (Optional) -->
547      <div class="mb-2">
548          <label for="shop_address2">Address 2 <span class="text-muted">(Optional)</span></label>
549          <input type="text" class="form-control" id="shop_address2" name="shop_address2" placeholder="Apartment or Suite">
550      </div> <!-- END - Address 2 (Optional) -->
551
552      <!-- Dealership Information Row 5-->
553      <div class="row justify-content-center">
554          <!-- Country -->
555          <div class="col-md-4 mb-3">
556              <label for="shop_country">Country</label>
557              <select class="custom-select d-block w-100" id="shop_country" name="shop_country" required>
558                  <option value="">Choose...</option>
559                  <option value="CAN" selected>Canada</option>
560              </select>
561              <div class="invalid-feedback">
562                  Please select a valid country.
563              </div>
564          </div> <!-- END - Country -->
565
566          <!-- Province -->
567          <div class="col-md-4 mb-3">
568              <label for="shop_province">Province/Territory:</label>
569              <select class="custom-select d-block w-100" id="shop_province" name="shop_province" required>
570                  <option value="">Choose...</option>
571                  <option value="AB">Alberta</option>
572                  <option value="BC">British Columbia</option>
573                  <option value="MB">Manitoba</option>
574                  <option value="NB">New Brunswick</option>
575                  <option value="NL">Newfoundland and Labrador</option>
576                  <option value="NT">Northwest Territories</option>
577                  <option value="NS">Nova Scotia</option>
578                  <option value="NU">Nunavut</option>
579                  <option value="ON">Ontario</option>
580                  <option value="PE">Prince Edward Island</option>
581                  <option value="QC">Quebec</option>
582                  <option value="SK">Saskatchewan</option>
583                  <option value="YT">Yukon</option>
584              </select>
585              <div class="invalid-feedback">
586                  Please provide a valid province or territory.
587              </div>
588          </div> <!-- END - Province -->
589
590          <!-- Postal Code -->
591          <div class="col-md-4 mb-3">
592              <label for="shop_postalcode">Postal Code:</label>
593              <input type="text" class="form-control" id="shop_postalcode" name="shop_postalcode" placeholder="A1A 1A1" required>
594              <div class="invalid-feedback">
595                  Postal code required.
596              </div>
597          </div>
598
599      <script>
600          $(document).ready(function() {
601              $('#shop_postalcode').on('input', function() {
602                  var val = $(this).val();
603                  val = val.replace(/[^A-Za-z0-9]/g, '');
604                  if (val.length > 3 && val.length <= 6) {
605                      val = val.substring(0, 3) + ' ' + val.substring(3);
606                  } else if (val.length > 6) {
607                      val = val.substring(0, 3) + ' ' + val.substring(3, 6);
608                  }
609                  $(this).val(val.toUpperCase());
610              });
611          });
612      </script> <!-- END -Postal Code -->

```

```

621 |      <!-- Customer Information -->
622 |      <h4 class="mb-3 text-center">Customer Information</h4>
623 |
624 |      <!-- Customer Information Row 1 -->
625 |      <div class="row justify-content-center">
626 |
627 |          <!-- First Name -->
628 |          <div class="col-md-2 mb-3">
629 |              <label for="cust_fname">First Name</label>
630 |              <input type="text" class="form-control" id="cust_fname" name="cust_fname" placeholder="" required>
631 |              <div class="invalid-feedback">
632 |                  First name required.
633 |              </div>
634 |          </div> <!-- END - First Name -->
635 |
636 |          <!-- Last Name -->
637 |          <div class="col-md-2 mb-3">
638 |              <label for="cust_lname"> Last Name</label>
639 |              <input type="text" class="form-control" id="cust_lname" name="cust_lname" placeholder="" required>
640 |              <div class="invalid-feedback">
641 |                  Last name required.
642 |              </div>
643 |          </div> <!-- END - Last Name -->
644 |
645 |          <!-- Policy Type -->
646 |          <div class="col-md-2 mb-3">
647 |              <label for="policy_type">Policy Type</label>
648 |              <input type="text" class="form-control" id="policy_type" name="policy_type" placeholder="" required>
649 |              <div class="invalid-feedback">
650 |                  Policy type required.
651 |              </div>
652 |          </div> <!-- END - Policy Type -->
653 |
654 |          <!-- Coverage Type -->
655 |          <div class="col-md-2 mb-3">
656 |              <label for="coverage_type">Coverage Type</label>
657 |              <input type="text" class="form-control" id="coverage_type" name="coverage_type" placeholder="" required>
658 |              <div class="invalid-feedback">
659 |                  Coverage type required.
660 |              </div>
661 |          </div> <!-- END - Coverage Type -->
662 |
663 |      </div> <!-- END - Customer Information Row 1 -->
664 |
665 |      <!-- Customer Information Row 2 -->
666 |      <div class="row justify-content-center">
667 |          <!-- Vehicle VIN -->
668 |          <div class="col-md-2 mb-3">
669 |              <label for="vin">Vehicle VIN</label>
670 |              <input type="text" class="form-control" id="vin" name="vin" placeholder="" required>
671 |              <div class="invalid-feedback">
672 |                  Valid VIN required.
673 |              </div>
674 |          </div> <!-- END - Vehicle VIN -->
675 |
676 |          <!-- Vehicle Make (Future Feature) -->
677 |          <div class="col-md-2 mb-3">
678 |              <label for="make">Make</label>
679 |              <input type="text" class="form-control" id="make" name="make" placeholder="(Future Feature)" readonly>
680 |          </div> <!-- END - Vehicle Make (Future Feature) -->
681 |
682 |          <!-- Vehicle Model (Future Feature) -->
683 |          <div class="col-md-2 mb-3">
684 |              <label for="model">Model</label>
685 |              <input type="text" class="form-control" id="model" name="model" placeholder="(Future Feature)" readonly>
686 |          </div> <!-- END - Vehicle Model (Future Feature) -->
687 |

```

```

688     <!-- Vehicle Year (Future Feature) -->
689     <div class="col-md-2 mb-3">
690       <label for="veh_year">Year</label>
691       <input type="text" class="form-control" id="veh_year" name="veh_year" placeholder="(Future Feature)" readonly>
692     </div> <!-- END - Vehicle Year (Future Feature) -->
693   </div> <!-- END - Customer Information Row 2 -->
694
695   <!-- Vehicle VIN Validation-->
696   <script>
697     $(document).ready(function() {
698       $('#vin').on('input', function() {
699         var val = $(this).val().toUpperCase();
700         val = val.replace(/\D/g, '');
701         if (val.length == 17) {
702           $.get('{% url "web_appvalidate_vin" %}', {vin: val}, function(response) {
703             if (response.valid) {
704               $('#make').val(response.make);
705               $('#model').val(response.model);
706               $('#veh_year').val(response.year);
707             } else {
708               $('#make').val('');
709               $('#model').val('');
710               $('#veh_year').val('');
711             }
712             $('#vin').removeClass('is-invalid');
713             $('#vin').addClass('is-valid');
714           }, fail(function() {
715             $('#make').val('');
716             $('#model').val('');
717             $('#veh_year').val('');
718             $('#vin').removeClass('is-valid');
719             $('#vin').addClass('is-invalid');
720           });
721         } else {
722           $('#make').val('');
723           $('#model').val('');
724           $('#veh_year').val('');
725           $('#vin').removeClass('is-valid');
726           $('#vin').addClass('is-invalid');
727         }
728         $(this).val(val);
729       });
730
731       $('#vin').on('blur', function() {
732         if ($(this).val().length != 17) {
733           $(this).removeClass('is-valid');
734           $(this).addClass('is-invalid');
735         } else {
736           $(this).removeClass('is-invalid');
737           $(this).addClass('is-valid');
738         }
739       });
740     });
741   </script> <!-- END - Vehicle VIN Validation-->
742
743   <!-- Problem Information Row -->
744   <div class="row justify-content-between align-items-center mb-3">
745     <div class="col text-center">
746       <h4 class="mb-2">Problem</h4>
747     </div>
748
749   </div>
750
751
752   <div class="problem-row-container justify-content-center">
753     <div class="row justify-content-center problem-row" id="problemrow">
754       <div class="col-md-10 mb-4">
755         <div class="form-group">
756           <div class="row justify-content-center">
757             <!-- Type -->
758             <div class="col-md-2 mb-4">
759               <label for="prob_type">Type</label>
760               <select class="custom-select d-block w-100" id="prob_type" name="prob_type" required>
761                 <option value="">Choose...</option>
762                 <option value="prob_fabric">Fabric</option>
763                 <option value="prob_leathvin">Leather/Vinyl</option>
764                 <option value="prob_paint">Paint</option>
765                 <option value="prob_rust">Rust</option>
766                 <option value="prob_theft">Theft</option>
767                 <option value="prob_undercoat">Undercoat</option>
768               </select>
769               <div class="invalid-feedback">
770                 Please select a type.
771               </div>
772             </div> <!-- END - Type -->
773
774             <!-- Description -->
775             <div class="col-md-2 mb-4">
776               <label for="prob_description">Description</label>
777               <input type="text" class="form-control" id="prob_description" name="prob_description" placeholder="" required>
778               <div class="invalid-feedback">
779                 Description required.
780               </div>
781             </div> <!-- END - Description -->
782
783             <!-- Part -->
784             <div class="col-md-2 mb-4">
785               <label for="prob_part">Part</label>
786               <input type="text" class="form-control" id="prob_part" name="prob_part" placeholder="" required>
787               <div class="invalid-feedback">
788                 Part required.
789               </div>
790             </div> <!-- END - Part -->
791
792             <!-- Side -->
793             <div class="col-md-2 mb-4">
794               <label for="prob_side">Side</label>
795               <select class="custom-select d-block w-100" id="prob_side" name="prob_side" required>
796                 <option value="">Choose...</option>
797                 <option value="prob_bothsides">Both</option>
798                 <option value="prob_leftside">Left</option>
799                 <option value="prob_rightside">Right</option>
800                 <option value="prob_noside">N/A</option>
801               </select>
802               <div class="invalid-feedback">
803                 Please select a side.
804               </div>
805             </div> <!-- END - Side -->
806
807
808
809
810
811

```

```

811      <!-- Goodwill -->
812      <div class="col-md-2 mb-4">
813        <label for="goodwill">Goodwill</label>
814        <div class="input-group">
815          <div class="input-group-prepend">
816            <span class="input-group-text">$</span>
817          </div>
818          <input type="number" class="form-control goodwill" id="goodwill" name="goodwill" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
819        </div>
820      </div>
821
822      <script>
823        function roundToTwoDecimalPlaces(inputField) {
824          inputField.value = parseFloat(inputField.value).toFixed(2);
825        }
826      </script> <!-- END - Goodwill -->
827
828      <!-- Remove/Add Problem Buttons -->
829      <div class="col-md-2">
830        <div class="col-md-12 mb-4 d-flex justify-content-end align-items-center">
831          <!-- Remove Problem Button -->
832          <button type="button" onclick="removeRow(this)" class="btn btn-dark remove-problem-btn" style="margin-top: 30px; width: 35px; margin-right: 10px;"></button>
833
834          <!-- Add Another Problem Button -->
835          <button type="button" onclick="duplicate()" class="btn btn-dark" style="margin-top: 30px; width: 35px;" id="add-problem-btn"></button>
836        </div>
837      </div> <!-- END - Remove/Add Problem Buttons -->
838
839      <!-- Add Problem Rows -->
840      <script src="https://code.jquery.com/jquery-3.6.3.js"></script>
841
842      $(document).ready(function() {
843        // Add new row on button click
844        $('.problem-row-container').on('click', '#add-problem-btn', function(){
845          var newRow = $('.problem-row-list').clone();
846          newRow.find('input[type=text]').val(""); // clear input fields
847          newRow.find('select').prop('selectedIndex', 0); // reset select fields
848
849          // Add Bootstrap classes to cloned elements
850          newRow.find('.col-md-2').removeClass('col-md-2').addClass('col-md-2 mb-3');
851          newRow.find('.custom-select').removeClass('d-block w-100').addClass('form-control');
852
853          // Set Goodwill field value to $0.00
854          newRow.find('.goodwill').val('0.00');
855
856          newRow.attr('class', 'row justify-content-center problem-row'); // add class name to new row
857
858          // Insert new row after the clicked row
859          newRow.insertAfter($(this).closest('.problem-row'));
860        });
861      });
862    </script> <!-- END - Add Problem Rows -->
863
864    <!-- Remove row on button click -->
865    <script>
866      function removeRow(button) {
867        var row = button.closest('.problem-row');
868        var rows = document.querySelectorAll('.problem-row');
869        if (rows.length > 1) {
870          row.remove();
871        }
872      }
873    </script> <!-- END - Remove row on button click -->
874
875  </div>
876 </div>
877 </div>
878
879
880  <!-- Details Row -->
881  <h4 class="mb-3 text-center">Details</h4>
882  <div class="row justify-content-center">
883    <div class="col-12">
884      <textarea class="form-control" id="claim_details" name="claim_details" placeholder="Enter text here" rows="5"></textarea>
885    </div>
886  </div> <!-- END - Details Row -->
887
888  <br>
889  <br>
890
891  <!-- Upload Claim Images Row -->
892  <h4 class="mb-3 text-center">Upload Claim Images</h4>
893
894  <!-- File upload section -->
895  <div class="row justify-content-center">
896    <input type="file" id="file-input" onchange="handleFileSelect(event)" multiple>
897  </div>
898
899  <br>
900
901  <!-- Image preview section -->
902  <div id="preview">
903    <div class="preview-row"></div>
904  </div>

```

```

915      <script>
916          function handleFileSelect(event) {
917              const files = event.target.files;
918              const preview = document.getElementById("preview");
919              let counter = 0;
920
921              for (let i = 0; i < files.length; i++) {
922                  const file = files[i];
923                  const reader = new FileReader();
924
925                  reader.onload = function (event) {
926                      const img = new Image();
927                      img.src = event.target.result;
928
929                      img.onload = function () {
930                          // Resize the image to 250 x 250
931                          const canvas = document.createElement("canvas");
932                          const ctx = canvas.getContext("2d");
933
934                          const width = img.width;
935                          const height = img.height;
936                          const max_size = 250;
937
938                          let new_width;
939                          let new_height;
940
941                          if (width > height) {
942                              new_width = max_size;
943                              new_height = (height * max_size) / width;
944                          } else {
945                              new_height = max_size;
946                              new_width = (width * max_size) / height;
947                          }
948
949                          canvas.width = new_width;
950                          canvas.height = new_height;
951
952                          ctx.drawImage(img, 0, 0, new_width, new_height);
953
954                          const resizedImg = document.createElement("img");
955                          resizedImg.src = canvas.toDataURL();
956
957                          // Append the resized image to the corresponding row
958                          const rows = preview.querySelectorAll(".preview-row");
959                          const lastRow = rows[rows.length - 1];
960                          if (lastRow && lastRow.querySelectorAll(".preview-column").length < 3) {
961                              // Add the image to the last row if it has enough space
962                              const column = document.createElement("div");
963                              column.classList.add("preview-column");
964                              column.appendChild(resizedImg);
965                              lastRow.appendChild(column);
966                          } else {
967                              // Create a new row and add the image to it
968                              const row = document.createElement("div");
969                              row.classList.add("preview-row");
970                              const column = document.createElement("div");
971                              column.classList.add("preview-column");
972                              column.appendChild(resizedImg);
973                              row.appendChild(column);
974                              preview.appendChild(row);
975                          }

```

```

985 // Check if all images have been loaded
986 counter++;
987 if (counter === files.length) {
988 // Add a new row if necessary to make sure the last row is filled
989 if (counter % 3 !== 0) {
990 const lastRow = preview.querySelector(".preview-row:last-child");
991 const emptyColumns = 3 - (counter % 3);
992 for (let j = 0; j < emptyColumns; j++) {
993 const emptyColumn = document.createElement("div");
994 emptyColumn.classList.add("preview-column");
995 lastRow.appendChild(emptyColumn);
996 }
997 }
998 }
999
1000 }
1001 reader.readAsDataURL(file);
1002
1003 // Add a new row if necessary
1004 if (i % 3 === 0) {
1005 const row = document.createElement("div");
1006 row.classList.add("preview-row");
1007 preview.appendChild(row);
1008 }
1009
1010 }
1011 document.getElementById("claim_images").addEventListener("change", handleFileSelect);
1012
1013 </script>
1014
1015 <style>
1016 .preview-row {
1017   display: flex;
1018   flex-wrap: wrap;
1019   margin-bottom: 10px;
1020 }
1021 .preview-column {
1022   flex: 1 0 31%;
1023   margin-right: 2%;
1024   margin-bottom: 10px;
1025 }
1026 </style> <!-- END - Image preview section -->
1027 <!-- END - Upload Claim Images Row -->

```

```

<!-- Estimate Row -->
<div class="mb-3 text-center">Estimates</div>
<!-- Estimate Tier 2 Checkbox -->
<style>
  input[type="checkbox"]#est_tier2 {
    margin-left: 140px; /* Change the value as needed */
    vertical-align: left;
  }
</style>
<input type="checkbox" style="padding-left: 30px" id="est_tier2" name="est_tier2">
<label for="est_tier2">Claim Over $1500</label>
<!-- END - Estimate Tier 2 Checkbox -->
<!-- Estimate Tier 1 Row (under $1500) -->
<div class="row justify-content-center">
  <!-- Estimate 1 -->
  <div class="col-md-3 mb-3">
    <label for="claim_estimate1">Estimate #1</label>
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">$</span>
      </div>
      <input type="number" class="form-control claim_estimate1" id="claim_estimate1" name="claim_estimate1" min="0" step=".01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
    </div>
  </div>
  <script>
    function roundToTwoDecimalPlaces(inputField) {
      inputField.value = parseFloat(inputField.value).toFixed(2);
    }
  </script> <!-- END - Estimate 1 -->
<!-- Estimate 2 -->
<div class="col-md-3 mb-3">
  <label for="claim_estimate2">Estimate #2</label>
  <div class="input-group">
    <div class="input-group-prepend">
      <span class="input-group-text">$</span>
    </div>
    <input type="number" class="form-control claim_estimate2" id="claim_estimate2" name="claim_estimate2" min="0" step=".01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
  </div>
  <script>
    function roundToTwoDecimalPlaces(inputField) {
      inputField.value = parseFloat(inputField.value).toFixed(2);
    }
  </script> <!-- END - Estimate 2 -->
<!-- Requested -->
<div class="col-md-3 mb-3">
  <label for="claim_req_amt">Requested</label>
  <div class="input-group">
    <div class="input-group-prepend">
      <span class="input-group-text">$</span>
    </div>
    <input type="number" class="form-control claim_req_amt" id="claim_req_amt" name="claim_req_amt" min="0" step=".01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
  </div>

```

```

1696     <script>
1697       function roundToTwoDecimalPlaces(inputField) {
1698         inputField.value = parseFloat(inputField.value).toFixed(2);
1699       }
1700     </script> <!-- END - Requested -->
1701   </div> <!-- END - Estimate Tier 1 Row (under $1500) -->
1702
1703   <!-- Hide Tier 2 Row (hidden unless claim over $1500 checkbox clicked) -->
1704   <script>
1705     $(document).ready(function() {
1706       // hide the Estimate Tier 2 section initially
1707       $('#west-tier2-section').hide();
1708
1709       // add an event listener to the Status field
1710       $('#west-tier2').on('change', function() {
1711         if ($('#this').is(':checked')) {
1712           $('#west-tier2-section').show();
1713         } else {
1714           $('#west-tier2-section').hide();
1715         }
1716       });
1717     });
1718   </script> <!-- END - Hide Tier 2 -->
1719
1720   <!-- Estimate Tier 2 Row (over $1500) -->
1721   <div class="row justify-content-center" id="est-tier2-section">
1722     <!-- Estimate 3 -->
1723     <div class="col-md-3 mb-3">
1724       label for="claim_estimates" Estimate #3:</label>
1725       <div class="input-group">
1726         <div class="input-group-prepend">
1727           <span class="input-group-text">$</span>
1728         </div>
1729         <input type="number" class="form-control claim_estimate" id="claim_estimate" name="claim_estimate" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1730       </div>
1731
1732       <script>
1733         function roundToTwoDecimalPlaces(inputField) {
1734           inputField.value = parseFloat(inputField.value).toFixed(2);
1735         }
1736       </script> <!-- END - Estimate 3 -->
1737
1738     <!-- Estimate 4 -->
1739     <div class="col-md-3 mb-3">
1740       label for="claim_estimated4" Estimate #4:</label>
1741       <div class="input-group">
1742         <div class="input-group-prepend">
1743           <span class="input-group-text">$</span>
1744         </div>
1745         <input type="number" class="form-control claim_estimated4" id="claim_estimated4" name="claim_estimated4" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1746       </div>
1747     </div>
1748
1749     <script>
1750       function roundToTwoDecimalPlaces(inputField) {
1751         inputField.value = parseFloat(inputField.value).toFixed(2);
1752       }
1753     </script> <!-- END - Estimate 4 -->
1754
1755     <!-- Estimate 5 -->
1756     <div class="col-md-3 mb-3">
1757       label for="claim_estimated5" Estimate #5:</label>
1758       <div class="input-group">
1759         <div class="input-group-prepend">
1760           <span class="input-group-text">$</span>
1761         </div>
1762         <input type="number" class="form-control claim_estimated5" id="claim_estimated5" name="claim_estimated5" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1763       </div>
1764     </div>
1765
1766     <script>
1767       function roundToTwoDecimalPlaces(inputField) {
1768         inputField.value = parseFloat(inputField.value).toFixed(2);
1769       }
1770     </script> <!-- END - Estimate 5 -->
1771
1772   </div> <!-- END - Estimate Tier 2 Row (over $1500) -->
1773
1774   <br>
1775
1776   <!-- Amount Row -->
1777   <div class="mb-3 text-center" style="text-align: center;">
1778     <h4>Actual Claim Amount</h4>
1779   </div>
1780
1781
1782   <!-- Agreed To -->
1783   <div class="row justify-content-center" style="text-align: center;">
1784     <!-- Agreed To -->
1785     <div class="col-md-3 mb-3">
1786       label for="claim_ant_agreed" Agreed Amount:</label>
1787       <div class="input-group">
1788         <div class="input-group-prepend">
1789           <span class="input-group-text">$</span>
1790         </div>
1791         <input type="number" class="form-control claim_ant_agreed" id="claim_ant_agreed" name="claim_ant_agreed" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1792       </div>
1793     </div>
1794
1795     <script>
1796       function roundToTwoDecimalPlaces(inputField) {
1797         inputField.value = parseFloat(inputField.value).toFixed(2);
1798       }
1799     </script> <!-- END - Agreed To -->
1800
1801     <!-- Deductible -->
1802     <div class="col-md-3 mb-3">
1803       label for="claim_deductible" Deductible:</label>
1804       <div class="input-group">
1805         <div class="input-group-prepend">
1806           <span class="input-group-text">$</span>
1807         </div>
1808         <input type="number" class="form-control claim_deductible" id="claim_deductible" name="claim_deductible" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1809       </div>
1810     </div>
1811
1812     <script>
1813       function roundToTwoDecimalPlaces(inputField) {
1814         inputField.value = parseFloat(inputField.value).toFixed(2);
1815       }
1816     </script> <!-- END - Deductible -->

```

```

1224      <!-- Net Amount -->
1225      <div class="col-md-3 mb-3">
1226          <label for="claim_net_amt">Net Amount</label>
1227          <div class="input-group">
1228              <div class="input-group-prepend">
1229                  <span class="input-group-text">$</span>
1230              </div>
1231              <input type="text" class="form-control claim_net_amt" id="claim_net_amt" name="claim_net_amt" value="0.00" disabled>
1232          </div>
1233
1234
1235      <script>
1236          // Get the input elements
1237          const agreedToInput = document.querySelector('.claim_amt_agreed');
1238          const deductibleInput = document.querySelector('.claim_deductible');
1239          const netAmountInput = document.querySelector('.claim_net_amt');
1240
1241          // Add event listener to the agreed to and deductible inputs
1242          agreedToInput.addEventListener('input', calculateNetAmount);
1243          deductibleInput.addEventListener('input', calculateNetAmount);
1244
1245          // Function to calculate net amount
1246          function calculateNetAmount() {
1247              const agreedTo = parseFloat(agreedToInput.value.replace('$', '')) || 0;
1248              const deductible = parseFloat(deductibleInput.value.replace('$', '')) || 0;
1249              const netAmount = agreedTo - deductible;
1250              netAmountInput.value = $(netAmount.toFixed(2));
1251          }
1252      </script> <!-- END - Net Amount -->
1253
1254  </div> <!-- END - Amount Row -->
1255
1256  <br>
1257  <br>
1258
1259
1260  <!-- Payment Row -->
1261  <h4 class="mb-3 text-center">Payment Information</h4>
1262
1263  <div class="row justify-content-center">
1264
1265      <!-- Cheque Number -->
1266      <div class="col-md-3 mb-3">
1267          <label for="cheque_num">Cheque #</label>
1268          <input type="text" class="form-control" id="cheque_num" name="cheque_num" placeholder="">
1269          <div class="invalid-feedback">
1270              Cheque number required.
1271          </div>
1272      </div> <!-- END - Cheque Number -->
1273
1274      <!-- Pay To -->
1275      <div class="col-md-3 mb-3">
1276          <label for="cheque_pay_to">Pay To</label>
1277          <input type="text" class="form-control" id="cheque_pay_to" name="cheque_pay_to" placeholder="">
1278          <div class="invalid-feedback">
1279              Name required.
1280          </div>
1281      </div> <!-- END - Pay To -->
1282
1283
1284  <!-- Include the Bootstrap and jQuery libraries -->
1285  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/css/bootstrap-datepicker.min.css">
1286  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
1287  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
1288  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
1289
1290  <!-- Include the Bootstrap datepicker library -->
1291  <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/js/bootstrap-datepicker.min.js"></script>
1292
1293  <!-- Cheque Date -->
1294  <div class="col-md-3 mb-3">
1295      <label for="cheque_date">Cheque Date:</label>
1296      <div class="input-group">
1297          <input type="text" class="form-control" id="cheque_date" name="cheque_date" value="{{ formatted_date }}>
1298          <div class="input-group-append">
1299              <span class="input-group-text"><i class="fa fa-calendar"></i></span>
1300          </div>
1301      </div>
1302
1303      <script>
1304          // Get the current date
1305          var currentDate = new Date();
1306
1307          // Set the value of the claim date field to the formatted date
1308          var options = { month: 'long', day: 'numeric', year: 'numeric' };
1309          var formattedDate = currentDate.toLocaleDateString('en-US', options);
1310          document.getElementById('cheque_date').value = formattedDate;
1311
1312          // Initialize the datepicker
1313          $('#cheque_date').datepicker({
1314              format: 'MM dd, yyyy',
1315              todayHighlight: true,
1316              autoclose: true,
1317              orientation: 'bottom',
1318              templates: {
1319                  leftArrow: '<i class="fa fa-angle-left"></i>',
1320                  rightArrow: '<i class="fa fa-angle-right"></i>'
1321              }
1322          });
1323      </script> <!-- END - Cheque Date -->
1324
1325      <!-- Cheque Amount -->
1326      <div class="col-md-3 mb-3">
1327          <label for="cheque_amt">Cheque Amount:</label>
1328          <div class="input-group">
1329              <div class="input-group-prepend">
1330                  <span class="input-group-text">$</span>
1331              </div>
1332              <input type="number" class="form-control cheque_amt" id="cheque_amt" name="cheque_amt" min="0" step="0.01" placeholder="0.00" value="0.00" required onblur="roundToTwoDecimalPlaces(this)">
1333          </div>
1334
1335      <script>
1336          function roundToTwoDecimalPlaces(inputField) {
1337              inputField.value = parseFloat(inputField.value).toFixed(2);
1338          }
1339      </script> <!-- END - Cheque Amount -->
1340
1341  </div> <!-- END - Payment Row -->

```

```

1349 |     <!-- Enable enter key to tab through entries -->
1350 |     <script>
1351 |         // Get all the input fields
1352 |         const inputFields = document.querySelectorAll('input, select');
1353 |
1354 |         // Add event listener to prevent default behavior when "Enter" key is pressed
1355 |         inputFields.forEach(function(inputField) {
1356 |             inputField.addEventListener('keydown', function(event) {
1357 |                 if (event.key === 'Enter') {
1358 |                     event.preventDefault();
1359 |                 }
1360 |             });
1361 |         });
1362 |     </script>
1363 |
1364 |     <script>
1365 |         const form = document.querySelector('form');
1366 |         const fields = form.querySelectorAll('input[type="text"]');
1367 |
1368 |         fields.forEach((field, index) => {
1369 |             field.addEventListener('keydown', (event) => {
1370 |                 if (event.key === 'Enter') {
1371 |                     event.preventDefault();
1372 |                     fields[index+1].focus();
1373 |                 }
1374 |             });
1375 |         });
1376 |     </script> <!-- END - Enable enter key to tab through entries -->
1377 |
1378 |     <!-- Prevent form submission when required fields are not filled in -->
1379 |     <script>
1380 |         const myForm = document.getElementById('claimForm');
1381 |         claimForm.addEventListener('submit', (event) => {
1382 |             event.preventDefault();
1383 |         });
1384 |     </script>
1385 |
1386 |     <script>
1387 |         // Disabling form submissions if there are invalid fields
1388 |         (function() {
1389 |             'use strict';
1390 |
1391 |             window.addEventListener('load', function() {
1392 |                 // Fetch all the forms we want to apply custom Bootstrap validation styles to
1393 |                 var forms = document.getElementsByClassName('needs-validation');
1394 |
1395 |                 // Loop over them and prevent submission
1396 |                 var validation = Array.prototype.filter.call(forms, function(form) {
1397 |                     form.addEventListener('submit', function(event) {
1398 |                         if (form.checkValidity() === false) {
1399 |                             event.preventDefault();
1400 |                             event.stopPropagation();
1401 |                         } else {
1402 |                             // Submit the form
1403 |                             form.submit();
1404 |                         }
1405 |                         form.classList.add('was-validated');
1406 |                     }, false);
1407 |                 });
1408 |             }, false);
1409 |         })();
1410 |     </script> <!-- END - Prevent form submission when required fields are not filled in -->
1411 |
1412 |     <hr class="mb-4" > <!-- Divider Line -->
1413 |
1414 |     <!-- Submit Button -->
1415 |     <button class="btn btn-dark btn-lg btn-block" type="submit" id="claim_submitted" name="claim_submitted">Submit</button>

```

```

1417 |     <!-- Footer -->
1418 |     <footer class="my-5 pt-5 text-muted text-center text-small">
1419 |         <br>
1420 |     </footer>
1421 |     </form> <!-- END - Form -->
1422 |     </div> <!-- END - Form Format -->
1423 | </div> <!-- END - New Claim Form Class -->
1424 |
1425 | {% endblock content %}

```

## 4.2.2 views.py

```

25 def newclaim(request):
26     """Add a new claim."""
27     num_claims = Claim.objects.aggregate(num=Count('claim_number'))['num']
28     claim_number = num_claims + 1
29
30     #latest_claim = Claim.objects.last()
31     #claim_number = 1 if latest_claim is None else latest_claim.claim_number + 1
32     problem_form = inlineformset_factory(Claim, Problem, formset=ProblemForm, fields=('prob_type', 'prob_description', 'prob_part', 'prob_side', 'goodwill'))
33
34     if request.method != 'POST':
35         # No data submitted; create a blank form.
36         claim_form = ClaimForm()
37         problem_form = problem_form(instance=Claim())
38
39     else:
40         # POST data submitted; process data.
41         claim_form = ClaimForm(data=request.POST)
42         problem_form = problem_form(data=request.POST)
43
44         if claim_form.is_valid():
45             newclaim = claim_form.save(commit=False)
46             newclaim.claim_number = claim_number
47             if request.POST.get('status') == 'Approved':
48                 newclaim.approved = True
49             if request.POST.get('status') == 'Rejected':
50                 newclaim.rejected = True
51             newclaim.save()
52
53             problem_form.instance = newclaim
54             problem_form.save()
55
56             # Call the handle_uploaded_files function to save the uploaded images
57             files = request.FILES.getlist('claim_images')
58             file_paths = handle_uploaded_files(files)
59
60             # Get all existing claims and add the new claim to the queryset
61             claimlist = Claim.objects.order_by('date_added')
62             claimlist = list(claimlist)
63             claimlist.append(newclaim)
64
65             # DEBUGGING
66             print(f"New claim {newclaim} saved successfully!")
67             print(f"File paths: {file_paths}")
68
69             return redirect('web_app:claims')
70
71         # DEBUGGING
72         else:
73             print(claim_form.errors)
74
75         # Display a blank or invalid form.
76         context = {'claim_form': claim_form, 'problem_form': problem_form, 'claim_number': claim_number}
77         return render(request, 'web_app/newclaim.html', context)
78
79 # Lookup vehicle information from VIN - currently not working
80 def validate_vin(request):
81     vin = request.GET.get('vin', '')
82     url = f'http://vpic.nhtsa.dot.gov/api/vehicles/DecodeVinValues/{vin}?format=json'
83     response = requests.get(url)
84     if response.status_code == 200:
85         data = response.json()['Results'][0]
86         make = data['Make']
87         model = data['Model']
88         year = data['ModelYear']
89         return JsonResponse({'valid': True, 'make': make, 'model': model, 'year': year})
90     else:
91         return JsonResponse({'valid': False})

```

### 4.2.3 models.py

```

1  from django.db import models
2  from django.utils import timezone
3
4  # Claim Model
5  class Claim(models.Model):
6      # Claim Information
7      claim_number = models.AutoField(primary_key=True, default=1)
8      date_added = models.DateTimeField(auto_now_add=True)
9
10     STATUS_CHOICES = [
11         ('status_info', 'Information Only'),
12         ('status_pending', 'Pending'),
13         ('status_submitted', 'Submitted'),
14         ('status_approved', 'Approved'),
15         ('status_rejected', 'Rejected'),
16         ('status_paid', 'Paid'),
17     ]
18     status = models.CharField(max_length=50, choices=STATUS_CHOICES, default='Information Only')
19
20     claim_date = models.DateTimeField(blank=True, default=timezone.now)
21     effective_date = models.DateTimeField(blank=True, default=timezone.now)
22     process_date = models.DateTimeField(blank=True, default=timezone.now)
23
24     po_num = models.CharField(max_length=200, default='N/A')
25     file_num = models.CharField(max_length=200, default='N/A')
26     agreement = models.CharField(max_length=200, default='N/A')
27
28     # Approval Information
29     claim_approvedby = models.CharField(max_length=50, blank=True)
30     claim_approvedto = models.CharField(max_length=50, blank=True)
31     claim_approved_date = models.DateTimeField(blank=True, null=True, default='')
32
33     # Rejected Information
34     claim_rejectedby = models.CharField(max_length=50, blank=True)
35     claim_rejection_date = models.DateTimeField(blank=True, null=True, default='')
36     claim_rejection_reason = models.CharField(max_length=1000, blank=True)
37
38     # Repair Shop Information
39     shop_active = models.BooleanField(default=False)
40     shop_name = models.CharField(max_length=200, default='N/A')
41     shop_phone = models.CharField(max_length=12, default='123-456-7890')
42     shop_contact = models.CharField(max_length=200, blank=True)
43     shop_email = models.EmailField(blank=True)
44     shop_address = models.CharField(max_length=200, default='1234 Main St')
45     shop_address2 = models.CharField(max_length=200, blank=True)
46
47     COUNTRY_CHOICES = [
48         ('CAN', 'Canada'),
49     ]
50     shop_country = models.CharField(max_length=50, choices=COUNTRY_CHOICES, default='CAN')
51
52     PROVINCE_CHOICES = [
53         ('AB', 'Alberta'),
54         ('BC', 'British Columbia'),
55         ('MB', 'Manitoba'),
56         ('NB', 'New Brunswick'),
57         ('NL', 'Newfoundland and Labrador'),
58         ('NT', 'Northwest Territories'),
59         ('NS', 'Nova Scotia'),
60         ('NU', 'Nunavut'),
61         ('ON', 'Ontario'),
62         ('PE', 'Prince Edward Island'),
63         ('QC', 'Quebec'),
64         ('SK', 'Saskatchewan'),
65         ('YT', 'Yukon'),
66     ]
67     shop_province = models.CharField(max_length=50, choices=PROVINCE_CHOICES, default='Ontario')

```

```

71     # Customer Information
72     cust_fname = models.CharField(max_length=30, default='N/A')
73     cust_lname = models.CharField(max_length=30, default='N/A')
74
75     policy_type = models.CharField(max_length=50, default='N/A')
76     coverage_type = models.CharField(max_length=50, default='N/A')
77     vin = models.CharField(max_length=30, default='JTHBP262495004947')
78
79     # Claim Details
80     claim_details = models.CharField(max_length=1000, blank=True)
81
82     # Estimate Information
83     claim_estimate1 = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
84     claim_estimate2 = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
85     claim_req_amt = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
86
87     est_tier2 = models.BooleanField(default=False)
88
89     claim_estimate3 = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
90     claim_estimate4 = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
91     claim_estimate5 = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
92
93
94     # Amount Information
95     claim_amt_agreed = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
96     claim_deductible = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
97
98     claim_net_amt = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True, default=0.00)
99
100    # Payment Information
101    cheque_num = models.CharField(max_length=50, blank=True)
102    cheque_pay_to = models.CharField(max_length=50, blank=True)
103    cheque_date = models.DateTimeField(blank=True, null=False, default='')
104    cheque_amt = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
105
106    def save(self, *args, **kwargs):
107        # Convert empty strings to None before saving
108        if not self.claim_estimate1:
109            self.claim_estimate1 = 0.00
110        if not self.claim_estimate2:
111            self.claim_estimate2 = 0.00
112        if not self.claim_estimate3:
113            self.claim_estimate3 = 0.00
114        if not self.claim_estimate4:
115            self.claim_estimate4 = 0.00
116        if not self.claim_estimate5:
117            self.claim_estimate5 = 0.00
118        if not self.claim_net_amt:
119            self.claim_net_amt = 0.00
120        if not self.cheque_amt:
121            self.cheque_amt = 0.00
122        super().save(*args, **kwargs)
123
124    def __str__(self):
125        return f"Claim {self.claim_number}"

```

### 4.2.3 forms.py

```

1  from django import forms
2  from .models import Claim, Problem
3  from django.forms import inlineformset_factory
4
5  class ClaimForm(forms.ModelForm):
6      claim_date = forms.DateTimeField(input_formats=['%B %d, %Y'])
7      problem_formset = inlineformset_factory(Claim, Problem, fields=('prob_type', 'prob_description', 'prob_part', 'prob_side', 'goodwill'))
8
9      class Meta:
10          model = Claim
11          fields = '__all__'
12
13      def __init__(self, *args, **kwargs):
14          super().__init__(*args, **kwargs)
15          self.fields['claim_approvedby'].required = False
16          self.fields['claim_approvedto'].required = False
17          self.fields['claim_approved_date'].required = False
18          self.fields['claim_rejectedby'].required = False
19          self.fields['claim_rejection_date'].required = False
20          self.fields['claim_rejection_reason'].required = False
21
22      def clean(self):
23          cleaned_data = super().clean()
24          status = cleaned_data.get('status')
25
26          if status == 'Approved':
27              if not cleaned_data.get('claim_approvedby'):
28                  self.add_error('claim_approvedby', 'This field is required.')
29              if not cleaned_data.get('claim_approvedto'):
30                  self.add_error('claim_approvedto', 'This field is required.')
31              if not cleaned_data.get('claim_approved_date'):
32                  self.add_error('claim_approved_date', 'This field is required.')
33
34          if status == 'Rejected':
35              if not cleaned_data.get('claim_rejectedby'):
36                  self.add_error('claim_rejectedby', 'This field is required.')
37              if not cleaned_data.get('claim_rejection_date'):
38                  self.add_error('claim_rejection_date', 'This field is required.')
39              if not cleaned_data.get('claim_rejection_reason'):
40                  self.add_error('claim_rejection_reason', 'This field is required.')
41
42          return cleaned_data

```

## 4.3 All Claims Webpage

### 4.3.1 All Claims Tab

Navigate to the All Claims tab to enter a new claim into the system. This is the default tab when navigating to the claims webpage from other webpages within the portal.



### 4.3.2 All Claims Default View

The default view in all claims includes the claim number, the date the claim was added, claim status, claim date, effective date, process date, PO number, file number, agreement number, shop name, customer first name, customer last name, policy type, coverage type, vehicle VIN number, and claim amount. There is also a search bar at the top which can be implemented in the future to search for claims via keywords.

Search (Future Feature)															Search
Show/Hide Additional Claim Information															
Claim #	Date Added	Status	Claim Date	Effective Date	Process Date	PO #	File #	Agreement #	Shop Name	Customer First Name	Customer Last Name	Policy Type	Coverage Type	Vehicle VIN	Claim Amount
1	March 25, 2023	Information Only	March 24, 2023	March 24, 2023	March 24, 2023	456	567	678	Maple Nissan	Harry	Kalaki	Interior	Upholstery Protection	1N4AL3AP1FN863724	
2	March 25, 2023	Rejected	March 24, 2023	March 24, 2023	March 24, 2023	144	87	213	Mississauga Hyundai	Johnny	Thompson	Interior	Upholstery Protection	KMHD4M41C36U25130	
3	March 25, 2023	Approved	March 20, 2023	March 22, 2023	March 23, 2023	3564	634	2342	Lindsay Kia	Bethany	Peters	Exterior	Paint Protection	KNDKNEH22A5050092	\$97.36
4	March 25, 2023	Pending	March 24, 2023	March 24, 2023	March 24, 2023	654	321	789	Alta Infiniti	Jessica	Lee	Exterior	Rust Protection	SN1AL0MMBEL549388	\$50.97
5	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	34523	2354	452	ToyotaTown	Barry	Louis	Interior	Fabric Protection	4TARN81P5PZ122020	
6	March 25, 2023	Paid	March 24, 2023	March 24, 2023	March 24, 2023	12344	3452	54757	Lexus of London	Sherry	Bennitt	Interior	Anti Theft Protection	JTHGL5EFXB5043474	\$126.84
7	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	18456	18456	8257	Mississauga Hyundai	Lisa	Wong	Interior	Undercoat Protection	5NMSG73DX9H317387	
8	March 25, 2023	Submitted	March 24, 2023	March 24, 2023	March 24, 2023	18456	9036	8257	Lindsay Kia	Valentina	Estrada	Interior	Fabric Protection	KNDJCT33645S91462	

#### **4.3.3 All Claims Expanded View**

To show all claim information, the user can click on the button. This button can be pressed again to toggle back to the default view.

#### 4.3.4 Claim Number Link

When a claim number is clicked, the user will be redirected to a new tab listing the claim's information. Only information filled out in a claim will be shown (blank sections are hidden). The following example shows Claim 1's information, showing only completed information (you can see that there are no estimates, claim amounts, payment information, etc.).

All Claims    New Claim    **View Claim 1**    Policies    Warranties

#### Claim 1

##### Claim Information

**Date Added:**  
March 25, 2023

**Status:**  
Information Only

**Claim Date:**  
March 25, 2023

**Claim Date:**  
March 24, 2023

**Effective Date:**  
March 24, 2023

**Process Date:**  
March 24, 2023

**PO #:**  
456

**File #:**  
567

**Agreement #:**  
678

##### Repair Shop Information

**Active:**  
YES

**Shop Name:**  
Maple Nissan

**Phone Number:**  
888-356-7509

**Address:**  
100 Auto Caughan Dr.

**Address 2:**  
Floor 2

**Country:**  
CAN

**Province/Territory:**  
ON

**Postal Code:**  
L6A 4A1

##### Customer Information

**First Name:**  
Harry

**Last Name:**  
Kalaki

**Policy Type:**  
Interior

**Coverage Type:**  
Upholstery Protection

**Vehicle VIN:**  
1N4AL3AP1FN863724

##### Problem

The left front driver's seat had some missing areas on the leather protection application.

## 4.4 All Claims Code

### 4.4.1 claims.html

```

1  (3 extends "Web_app/base.html" $)
2
3  <body>
4  ({ block content %}
5  <br>
6
7  <!-- Navigation Tabs, source: https://getbootstrap.com/docs/4.0/components/navbar/ -->
8  <ul class="nav nav-tabs">
9    <li class="nav-item" role="presentation">
10      <a href="#" class="nav-link active" id="allclaims-tab" data-bs-toggle="tab" href="{% url 'web_app:claim' %}" role="tab" aria-controls="nav-allclaims" aria-selected="true">All Claims</a>
11    </li>
12    <li class="nav-item" role="presentation">
13      <a href="#" class="nav-link" href="{% url 'web_app:newClaim' %}">New Claim</a>
14    </li>
15    <li class="nav-item" role="presentation">
16      <a href="#" class="nav-link disabled" href="#">Policies</a>
17    </li>
18    <li class="nav-item" role="presentation">
19      <a href="#" class="nav-link disabled" href="#">Warranties</a>
20    </li>
21  </ul>
22
23  <br>
24
25  <!-- List of Claims -->
26  <div style="margin-left:16px; margin-right:16px">
27
28    <!-- Search Bar -->
29    <br>
30    <form class="d-flex" role="search">
31      <input class="form-control me-2" type="search" placeholder="Search (Future Features)" aria-label="Search">
32      <button class="btn btn-outline-success" type="submit">Search</button>
33    </form>
34    <br>
35
36    <button id="toggle-button">Show/Hide Additional Claim Information</button>
37    <br>
38  </div>
39
40  <div class="table-wrapper">
41    <!-- Claims Table -->
42    <table style="padding-right: 8px" class="table">
43      <thead>
44        <tr>
45          <th style="vertical-align: top;">Claim #</th>
46          <th style="vertical-align: top;">Date Added</th>
47          <th style="vertical-align: top;">Status</th>
48          <th style="vertical-align: top;" class = "hidden" >Approved By</th>
49          <th style="vertical-align: top;" class = "hidden" >Approved To</th>
50          <th style="vertical-align: top;" class = "hidden" >Approved Date</th>
51          <th style="vertical-align: top;" class = "hidden" >Rejected By</th>
52          <th style="vertical-align: top;" class = "hidden" >Rejection Date</th>
53          <th style="vertical-align: top;" class = "hidden" >Rejection Reason</th>
54          <th style="vertical-align: top;">Claim Date</th>
55          <th style="vertical-align: top;">Effective Date</th>
56          <th style="vertical-align: top;">Process Date</th>
57          <th style="vertical-align: top;">PO #</th>
58          <th style="vertical-align: top;">File #</th>
59          <th style="vertical-align: top;">Agreement #</th>
60          <th style="vertical-align: top;" class = "hidden" >Shop Active</th>
61          <th style="vertical-align: top;">Shop Name</th>
62          <th style="vertical-align: top;" class = "hidden" >Shop Phone</th>
63          <th style="vertical-align: top;" class = "hidden" >Shop Contact</th>
64          <th style="vertical-align: top;" class = "hidden" >Shop Email</th>
65          <th style="vertical-align: top;" class = "hidden" >Shop Address</th>
66          <th style="vertical-align: top;" class = "hidden" >Shop Address 2</th>
67          <th style="vertical-align: top;" class = "hidden" >Shop Country</th>
68          <th style="vertical-align: top;" class = "hidden" >Shop Province/Territory</th>
69          <th style="vertical-align: top;" class = "hidden" >Shop Postal Code</th>
70          <th style="vertical-align: top;">Customer First Name</th>
71          <th style="vertical-align: top;">Customer Last Name</th>
72          <th style="vertical-align: top;">Policy Type</th>
73          <th style="vertical-align: top;">Coverage Type</th>
74          <th style="vertical-align: top;">Vehicle VIN</th>
75          <th style="vertical-align: top;" class = "hidden" >Details</th>
76          <th style="vertical-align: top;" class = "hidden" >Estimate #1</th>
77          <th style="vertical-align: top;" class = "hidden" >Estimate #2</th>
78          <th style="vertical-align: top;" class = "hidden" >Requested</th>
79          <th style="vertical-align: top;" class = "hidden" >Estimate #3</th>
80          <th style="vertical-align: top;" class = "hidden" >Estimate #4</th>
81          <th style="vertical-align: top;" class = "hidden" >Estimate #5</th>
82          <th style="vertical-align: top;" class = "hidden" >Agreed Amount</th>
83          <th style="vertical-align: top;" class = "hidden" >Deductible</th>
84          <th style="vertical-align: top;">Claim Amount</th>
85          <th style="vertical-align: top;" class = "hidden" >Cheque #</th>
86          <th style="vertical-align: top;" class = "hidden" >Cheque Pay To</th>
87          <th style="vertical-align: top;" class = "hidden" >Cheque Date</th>
88          <th style="vertical-align: top; padding-right: 30px;" class = "hidden" >Cheque Amount</th>
89        </tr>
90      </thead>

```

```

92     <tbody>
93         {% for claim in claims %}
94             <tr>
95                 
96                 <td>
97                     <a href="{% url 'web_app:viewclaim' claim.claim_number %}">{{ claim.claim_number }}</a>
98                     </td>
99
100                <td>
101                    | <div style="width: 100px; overflow-x: auto;">{{ claim.date_added|date:"F j, Y" }}</div>
102                </td>
103                (% if claim.status == "status_rejected" %)
104                <td style="color: #red">{{ claim.get_status_display }}</td>
105                (% elif claim.status == "status_paid" %)
106                <td style="color: #green">{{ claim.get_status_display }}</td>
107                (% else %)
108                <td>{{ claim.get_status_display }}</td>
109                (% endfor %)
110                | <td class="hidden">{{ claim.claim_approvedby|default:" " }}</td>
111                <td class="hidden">{{ claim.claim_approvedto|default:" " }}</td>
112                <td class="hidden">
113                    | <div style="width: 100px; overflow-x: auto;">% if claim.status != "status_approved" % (% else %){{ claim.claim_approved_date|date:"F j, Y" }}(% endif %)</div>
114                </td>
115                <td class="hidden">{{ claim.claim_rejectedby|default:" " }}</td>
116                <td style="width: 100px; overflow-x: auto;">% if claim.status != "status_rejected" % (% else %){{ claim.claim_rejection_date|date:"F j, Y" }}(% endif %)</div>
117                </td>
118                <td class="hidden">
119                    | <div style="width: 200px; overflow-x: auto;">{{ claim.claim_rejection_reason|default:" " }}</div>
120                </td>
121                <td>
122                    | <div style="width: 100px; overflow-x: auto;">{{ claim.claim_date|date:"F j, Y" }}</div>
123                </td>
124                <td>
125                    | <div style="width: 100px; overflow-x: auto;">{{ claim.effective_date|date:"F j, Y" }}</div>
126                </td>
127                <td>
128                    | <div style="width: 100px; overflow-x: auto;">{{ claim.process_date|date:"F j, Y" }}</div>
129                </td>
130                <td>
131                    | <div style="width: 100px; overflow-x: auto;">{{ claim.po_num }}</div>
132                </td>
133                <td>{{ claim.po_num }}</td>
134                <td>{{ claim_file_num }}</td>
135                <td>{{ claim_agreement }}</td>
136                <td class="hidden">(% if claim.shop_active %)YES(% else %)NO(% endif %)</td>
137                <td>{{ claim.shop_name }}</td>
138                <td class="hidden">
139                    | <div style="width: 100px; overflow-x: auto;">{{ claim.shop_phone }}</div>
140                </td>
141                <td class="hidden">{{ claim.shop_contact|default:" " }}</td>
142                <td class="hidden">{{ claim.shop_email|default:" " }}</td>
143                <td class="hidden">
144                    | <div style="width: 150px; overflow-x: auto;">{{ claim.shop_address }}</div>
145                </td>
146                <td class="hidden">{{ claim.shop_address2|default:" " }}</td>
147                <td class="hidden">{{ claim.shop_country }}</td>
148                <td class="hidden">{{ claim.shop_province }}</td>
149                <td class="hidden">
150                    | <div style="width: 100px; overflow-x: auto;">{{ claim.shop_postalcode }}</div>
151                </td>
152                <td>{{ claim.cust_name }}</td>
153                <td>{{ claim.cust_name }}</td>
154                <td>{{ claim.policy_type }}</td>
155                <td>{{ claim.coverage_type }}</td>
156                <td>{{ claim.vin }}</td>
157
158            <td class="hidden">
159                | <div style="width: 200px; overflow-x: auto;">{{ claim.claim_details }}</div>
160            </td>
161
162            <td class="hidden">(% if claim.Claim_estimate1 == 0 %) (% else %){{ claim.Claim_estimate1|default:" " }}(% endif %)</td>
163            <td class="hidden">(% if claim.Claim_estimate2 == 0 %) (% else %){{ claim.Claim_estimate2|default:" " }}(% endif %)</td>
164            <td class="hidden">(% if claim.Claim_req_amt == 0 %) (% else %){{ claim.Claim_req_amt|default:" " }}(% endif %)</td>
165            <td class="hidden">(% if claim.Claim_estimate3 == 0 %) (% else %){{ claim.Claim_estimate3|default:" " }}(% endif %)</td>
166            <td class="hidden">(% if claim.Claim_estimate4 == 0 %) (% else %){{ claim.Claim_estimate4|default:" " }}(% endif %)</td>
167            <td class="hidden">(% if claim.Claim_estimate5 == 0 %) (% else %){{ claim.Claim_estimate5|default:" " }}(% endif %)</td>
168            <td class="hidden">(% if claim.Claim_amt_agreed == 0 %) (% else %){{ claim.Claim_amt_agreed|default:" " }}(% endif %)</td>
169            <td class="hidden">(% if claim.Claim_deductible == 0 %) (% else %){{ claim.Claim_deductible|default:" " }}(% endif %)</td>
170            <td class="hidden">(% if claim.Claim_net_amt == 0 %) (% else %){{ claim.Claim_net_amt|default:" " }}(% endif %)</td>
171            <td class="hidden">{{ claim.cheque_num|default:" " }}</td>
172            <td class="hidden">{{ claim.cheque_pay_to|default:" " }}</td>
173            <td class="hidden">
174                | <div style="width: 100px; overflow-x: auto;">% if claim.cheque_amt == 0 % (% else %){{ claim.cheque_date|date:"F j, Y" }}(% endif %)</div>
175            </td>
176            <td style="padding-right: 30px;">{{ claim.cheque_amt }}</td>
177
178        <% empty %>
179        <tr>
180            | <td colspan="17">No claims have been added yet.</td>
181        </tr>
182        <% endfor %>
183    </tr>
184
185    <% empty %>
186    <tr>
187        | <td colspan="17">No claims have been added yet.</td>
188    </tr>
189
190<% endfor %>
191
192    </tbody>
193
194</table>
</div>

```

```

196  |    <!-- Show/Hide Additional Claim Information Script -->
197  |    <script>
198  |        // Get the button element
199  |        const toggleButton = document.getElementById("toggle-button");
200  |        const hiddenColumns = document.querySelectorAll('.hidden');
201  |        let columnsHidden = true;
202
203  |        // Add a click event listener to the button
204  |        toggleButton.addEventListener("click", function() {
205  |            // Toggle the boolean variable
206  |            columnsHidden = !columnsHidden;
207
208  |            // Loop through the hidden columns and toggle their display and class
209  |            for (let i = 0; i < hiddenColumns.length; i++) {
210  |                if (columnsHidden) {
211  |                    hiddenColumns[i].classList.add("hidden");
212  |                    hiddenColumns[i].setAttribute("hidden", "true");
213  |                } else {
214  |                    hiddenColumns[i].classList.remove("hidden");
215  |                    hiddenColumns[i].removeAttribute("hidden");
216  |                }
217  |            }
218  |        });
219  |    </script>
241  |    <style>
242  |        .table-wrapper {
243  |            padding-right: 8px;
244  |        }
245
246  |        table {
247  |            border-collapse: collapse;
248  |            width: 100%;
249  |            padding-right: 8px;
250  |        }
251  |        th, td {
252  |            padding: 8px;
253  |            text-align: center;
254  |            width: calc(100% / 16);
255  |            font-size: 14px;
256  |        }
257  |        th {
258  |            font-weight: bold;
259  |        }
260  |        .hidden {
261  |            display: none;
262  |            padding-right: 8px;
263  |        }
264
265  |    </style>
266
267  |    <br>
268
269  |</div>
270
271  |{% endblock content %}
272  |</body>

```

#### 4.4.2 viewclaim.html

```

1  {% extends 'web_app/base.html' %}
2  {% block content %}
3
4  <br>
5  <!-- Navigation Tabs, source: https://getbootstrap.com/docs/4.0/components/navbar/ -->
6
7  <ul class="nav nav-tabs">
8    <li class="nav-item" role="presentation">
9      <a class="nav-link" href="{% url 'web_app:claims'%}">All Claims</a>
10    </li>
11    <li class="nav-item" role="presentation">
12      <a class="nav-link" href="{% url 'web_app:newClaim'%}">New Claim</a>
13    </li>
14    <li class="nav-item" role="presentation">
15      <a class="nav-link active" href="{% url 'web_app:viewclaim' viewclaim.claim_number %}">View Claim {{ viewclaim.claim_number }}</a>
16    </li>
17    <li class="nav-item" role="presentation">
18      <a class="nav-link disabled" href="#">Policies</a>
19    </li>
20    <li class="nav-item" role="presentation">
21      <a class="nav-link disabled" href="#">Warranties</a>
22    </li>
23  </ul>
24
25  <div class="container">
26    <h1>Claim {{ viewclaim.claim_number }}</h1>
27    <h2 style="font-size: 25px; margin-top: 20px;">Claim Information</h2>
28    <br>
29
30    <h3 style="font-size: 21px;">Date Added:</h3>
31    <p>{{ viewclaim.date_added|date:"F j, Y" }}</p>
32    <h3 style="font-size: 21px;">Status:</h3>
33    {% if viewclaim.status == "status_rejected" %}
34      <p style="color: red">{{ viewclaim.get_status_display }}</p>
35    {% elif viewclaim.status == "status_paid" %}
36      <p style="color: green">{{ viewclaim.get_status_display }}</p>
37    {% else %}
38      <p>{{ viewclaim.get_status_display }}</p>
39    {% endif %}
40    <h3 style="font-size: 21px;">Claim Date:</h3>
41    <p>{{ viewclaim.date_added|date:"F j, Y" }}</p>
42
43    {% if viewclaim.status == "status_approved" %}
44      <h3 style="font-size: 21px;">Approved By:</h3>
45      <p>{{ viewclaim.claim_approvedby|default: " " }}</p>
46      <h3 style="font-size: 21px;">Approved To:</h3>
47      <p>{{ viewclaim.claim_approvedto|default: " " }}</p>
48      <h3 style="font-size: 21px;">Approved Date:</h3>
49      <p>{{ viewclaim.claim_approved_date|date:"F j, Y" }}</p>
50
51    {% elif viewclaim.status == "status_rejected" %}
52      <h3 style="font-size: 21px;">Rejected By:</h3>
53      <p>{{ viewclaim.claim_rejectedby|default: " " }}</p>
54      <h3 style="font-size: 21px;">Rejection Date:</h3>
55      <p>{{ viewclaim.claim_rejection_date|date:"F j, Y" }}</p>
56      <h3 style="font-size: 21px;">Reason for Rejection:</h3>
57      <p>{{ viewclaim.claim_rejection_reason|default: " " }}</p>
58
59    {% endif %}
60
61    <h3 style="font-size: 21px;">Claim Date:</h3>
62    <p>{{ viewclaim.claim_date|date:"F j, Y" }}</p>
63    <h3 style="font-size: 21px;">Effective Date:</h3>
64    <p>{{ viewclaim.effective_date|date:"F j, Y" }}</p>
65    <h3 style="font-size: 21px;">Process Date:</h3>
66    <p>{{ viewclaim.process_date|date:"F j, Y" }}</p>
67    <h3 style="font-size: 21px;">PO #:</h3>
68    <p>{{ viewclaim.po_num }}</td>
69    <h3 style="font-size: 21px;">File #:</h3>
70    <p>{{ viewclaim.file_num }}</td>
71    <h3 style="font-size: 21px;">Agreement #:</h3>
72    <p>{{ viewclaim.agreement }}</p>
73
74
75    <br>
76    <h2 style="font-size: 25px; margin-top: 20px;">Repair Shop Information</h2>
77    <br>
78    <h3 style="font-size: 21px;">Active:</h3>
79    <p>{{ viewclaim.shop_active }} YES {{ else }} NO {{ endif }}</p>
80    <h3 style="font-size: 21px;">Shop Name:</h3>
81    <p>{{ viewclaim.shop_name }}</p>
82    <h3 style="font-size: 21px;">Phone Number:</h3>
83    <p>{{ viewclaim.shop_phone }}</p>
84    <h3 style="font-size: 21px;">Address:</h3>
85    <p>{{ viewclaim.shop_contact }} {{ else }} {{ endif }}<h3 style="font-size: 21px;">Contact Name:</h3>{{ viewclaim.shop_contact|default: " " }}{{ endif }}</p>
86    <p>{{ viewclaim.shop_email }} {{ else }} {{ endif }}<h3 style="font-size: 21px;">Email:</h3>{{ viewclaim.shop_email|default: " " }}{{ endif }}</p>
87    <h3 style="font-size: 21px;">Address2:</h3>
88    <p>{{ viewclaim.shop_address2 }} {{ else }} {{ endif }}<h3 style="font-size: 21px;">Address:</h3>{{ viewclaim.shop_address2|default: " " }}{{ endif }}</p>
89    <h3 style="font-size: 21px;">Country:</h3>
90    <p>{{ viewclaim.shop_country }}</td>
91    <h3 style="font-size: 21px;">Province/Territory:</h3>
92    <p>{{ viewclaim.shop_province }}</td>
93    <h3 style="font-size: 21px;">Postal Code:</h3>
94    <p>{{ viewclaim.shop_postalcode }}</td>
95
96
97    <br>
98    <h2 style="font-size: 25px; margin-top: 20px;">Customer Information</h2>
99    <br>
100   <h3 style="font-size: 21px;">First Name:</h3>
101   <p>{{ viewclaim.cust_fname }}</p>
102   <h3 style="font-size: 21px;">Last Name:</h3>
103   <p>{{ viewclaim.cust_lname }}</p>
104   <h3 style="font-size: 21px;">Policy Type:</h3>
105   <p>{{ viewclaim.policy_type }}</p>
106   <h3 style="font-size: 21px;">Coverage Type:</h3>
107   <p>{{ viewclaim.coverage_type }}</p>
108   <h3 style="font-size: 21px;">Vehicle VIN:</h3>
109   <p>{{ viewclaim.vin }}</p>
110
111
112   <br>
113   <h2 style="font-size: 25px; margin-top: 20px;">Problem:</h2>
114   <br>
115   <p>{{ viewclaim.claim_details }}</td>
116   <br>

```

```

117     <div style="font-size: 25px; margin-top: 20px;">Estimates<br>
118     (% endif %)
119     <p>(% if viewclaim.claim_req_amt != 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimates</div>
120     <div style="font-size: 25px; margin-top: 20px; margin-left: 20px; border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">
121     <p>(% if viewclaim.claim_estimate1 == 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimate 1:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
122     <p>(% if viewclaim.claim_estimate2 == 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimate 2:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
123     <p>(% if viewclaim.claim_estimate3 == 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimate 3:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
124     <p>(% if viewclaim.claim_estimate4 == 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimate 4:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
125     <p>(% if viewclaim.claim_estimate5 == 0 %)<div style="font-size: 25px; margin-top: 20px;">Estimate 5:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
126     </div>
127     <p>(% if viewclaim.claim_amt_agreed == 0 %)<div style="font-size: 25px; margin-top: 20px;">Actual Claim Amount:</div>
128     <div style="font-size: 25px; margin-top: 20px; margin-left: 20px; border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">
129     <p>(% if viewclaim.claim_deductible == 0 %)<div style="font-size: 25px; margin-top: 20px;">Deductible:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
130     <p>(% if viewclaim.claim_net_amt == 0 %)<div style="font-size: 25px; margin-top: 20px;">Net Amount:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
131     </div>
132     <p>(% if viewclaim.cheque_num == "" %)<br><div style="font-size: 25px; margin-top: 20px;">Payment Information:</div>
133     <div style="font-size: 25px; margin-top: 20px; margin-left: 20px; border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">
134     <p>(% if viewclaim.cheque_pay_to == "0 %)"<div style="font-size: 25px; margin-top: 20px;">Pay To:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
135     <p>(% if viewclaim.cheque_date == "0 %)"<div style="font-size: 25px; margin-top: 20px;">Cheque Date:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
136     <p>(% if viewclaim.cheque_amt == 0 %)<div style="font-size: 25px; margin-top: 20px;">Cheque Amount:</div><div style="font-size: 25px; margin-top: 20px;">Default</div></p>
137     </div>
138     <br>
139   </div>
140   <br>
141   <br>
142 <% endblock %>
```

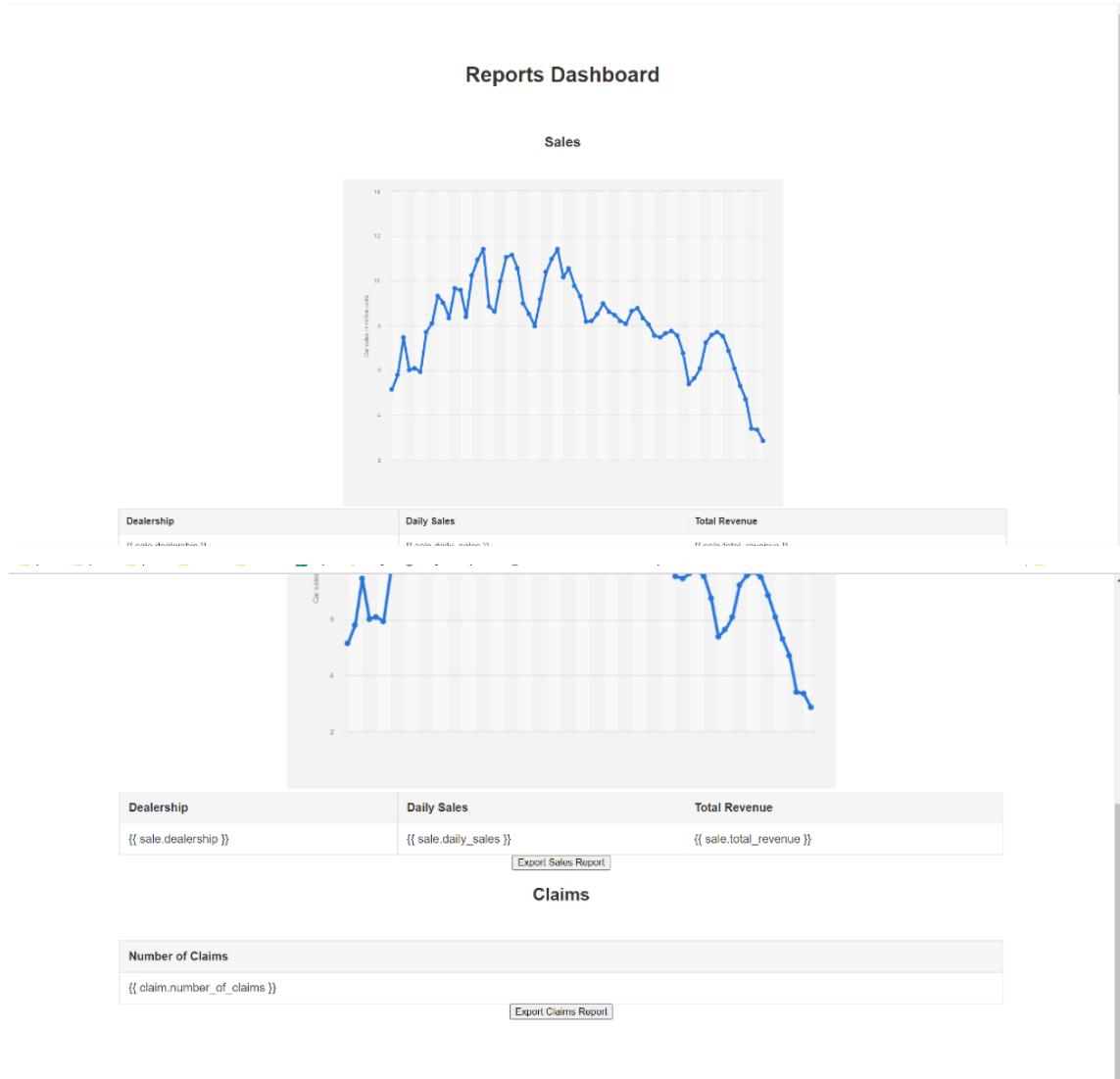
#### 4.4.3 views.py

```

18  def claims(request):
19      """show all claims"""
20      claims = Claim.objects.order_by('date_added')
21      context = {'claims': claims}
22      return render(request, 'web_app/claims.html', context)
...
137  # View Claim
138 def viewclaim(request, claim_number):
139     """Show a single Claim."""
140     # Make sure that the Work_order_form object exists in the database
141     singleClaim = get_object_or_404(Claim, claim_number=claim_number)
142
143     context = {'viewclaim': singleClaim}
144     return render(request, 'web_app/viewclaim.html', context)
```

## 5. Reports Dashboard

### 5.1 Web Page



\*In order to make the graphs interactive, JavaScript would be required. However, as this functionality falls outside the scope of this project at the moment, it is still under development.

\*\*The export button is currently under development as the database is still being developed. Once the database is complete, the export button will be implemented to allow users to export the data from the Sales Table and Claim Count Table to an CSV file.

## 5.2 HTML Code

File Edit Selection View Go Run Terminal Help reports\_dashboard.html - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```
< reports_dashboard.html >
C:\Users\kezhu\Downloads> reports_dashboard.html > ...
34     |     </tbody>
35     |   </table>
36     |   <div style="text-align: center;">
37     |     <form method="get" action="{% url 'export_sales_csv' %}">
38     |       <button type="submit">Export Sales Report</button>
39     |     </form>
40
41   <!-- Claims Table -->
42   <div class="section">
43     <h2 class="section-title">Claims</h2>
44
45     <table class="table">
46       <thead>
47         <tr>
48           <th>Number of Claims</th>
49         </tr>
50       </thead>
51       <tbody>
52         <tr>
53           <td>{{ claim.number_of_claims }}</td>
54         </tr>
55       </tbody>
56     </table>
57     <div style="text-align: center;">
58       <form method="get" action="{% url 'export_claims_csv' %}">
59         <button type="submit">Export Claims Report</button>
60       </form>
61     </div>
62   </div>
63 </body>
64 </html>
```

File Edit Selection View Go Run Terminal Help reports\_dashboard.html - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```
< reports_dashboard.html >
C:\Users\kezhu\Downloads> reports_dashboard.html > ...
1   <!DOCTYPE html>
2   <html>
3     <head>
4       <title>Analytics Reports</title>
5       <link rel="stylesheet" type="text/css" href="styles.css">
6     </head>
7
8     <body>
9       <h1 class="page-title">Reports Dashboard</h1>
10
11      <div class="section">
12        <h2 class="section-title">Sales</h2>
13
14        <div style="text-align: center;">
15          <div class="bar"></div>
16        </div>
17        <div style="text-align: center;">
18          
19        </div>
20
21        <table class="table">
22          <thead>
23            <tr>
24              <th>Dealership</th>
25              <th>Daily Sales</th>
26              <th>Total Revenue</th>
27            </tr>
28          </thead>
29          <tbody>
30            <tr>
31              <td>{{ sale.dealership }}</td>
32              <td>{{ sale.daily_sales }}</td>
33              <td>{{ sale.total_revenue }}</td>
34            </tr>
35          </tbody>
36        </table>
37
38    </div>
```

## 5.3 CSS Code

File Edit Selection View Go Run Terminal Help styles.css - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```
# styles.css ×
C:\> Users > kexzh > Downloads > # styles.css > body
34 .table th,
35 .table td {
36     padding: 12px;
37     border: 1px solid #ddd;
38     text-align: left;
39 }
40
41 .table th {
42     background-color: #f5f5f5;
43     font-weight: bold;
44 }
45
46 .button {
47     margin-top: 80px;
48     width: 200px;
49     font-family: 'DM Sans', sans-serif;
50     background-color: #000000;
51     color: white;
52     padding: 10px 20px;
53     border: none;
54     border-radius: 4px;
55     cursor: pointer;
56     text-align: center;
57     display: block;
58     margin: 0 auto;
59     text-decoration: none;
60 }
61
62 .button:hover {
63     background-color: #3e558e;
64 }
65
66 /* Define the container for the graph */
67 .graph-container {
68     width: 100%;
69     height: 300px;

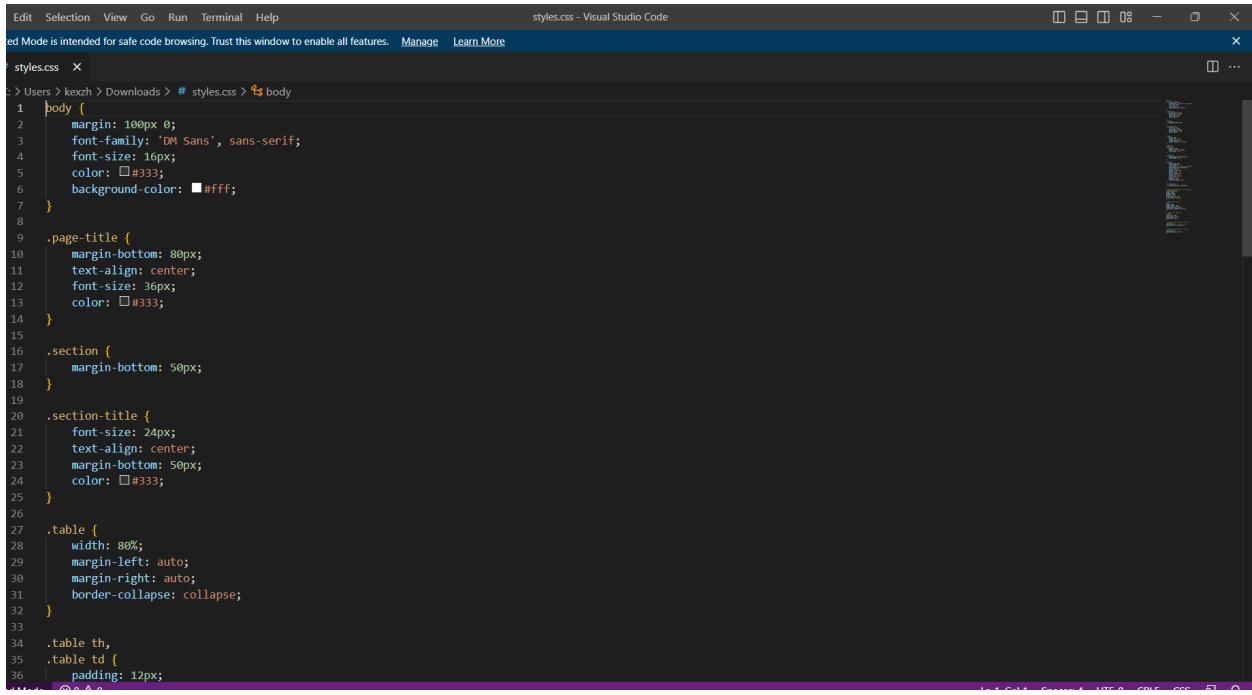
```

File Edit Selection View Go Run Terminal Help styles.css - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```
# styles.css ×
C:\> Users > kexzh > Downloads > # styles.css > body
0 .graph-container {
68     width: 100%;
69     height: 300px;
70     display: flex;
71     flex-direction: row;
72     align-items: flex-end;
73 }
74
75 /* Define the bars */
76 .bar {
77     width: 30px;
78     margin-right: 10px;
79     background-color: blue;
80     transition: height 0.5s ease;
81 }
82
83 /* Define the labels */
84 .label {
85     font-size: 12px;
86     text-align: center;
87     margin-top: 5px;
88 }
89
90 /* Define hover effect on bars */
91 .bar:hover {
92     background-color: darkblue;
93 }
94
95 /* Define active effect on bars */
96 .bar.active {
97     background-color: red;
98 }
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF CSS

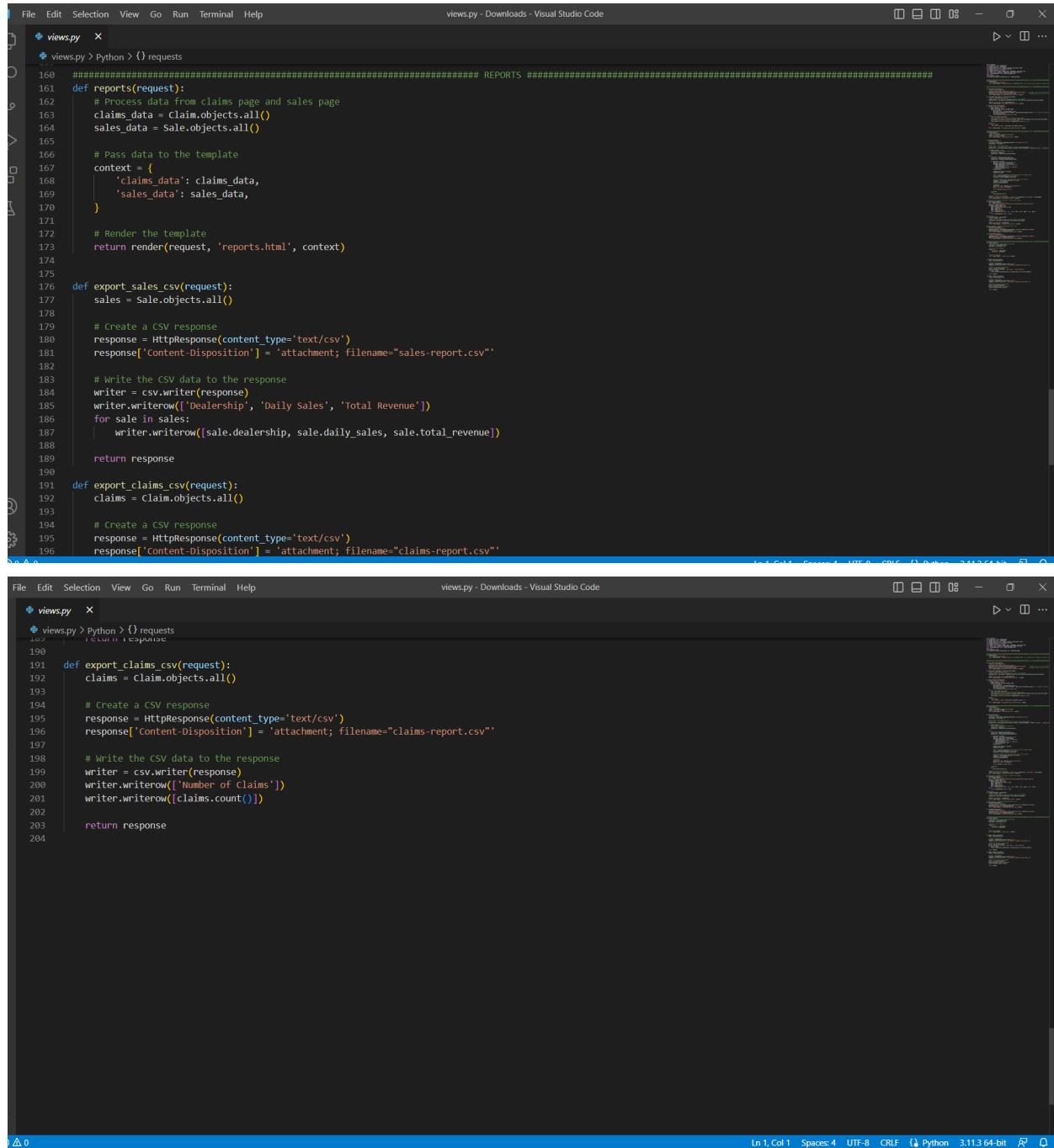


The screenshot shows a window of Visual Studio Code with the title "styles.css - Visual Studio Code". The menu bar includes "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". A status bar at the bottom shows "1 L 0 C 0 S 0 A 0 M 0 F 0 E 0". The code editor contains the following CSS code:

```
:> Users > kexzh > Downloads > # styles.css > ↵ body
1  body {
2    margin: 100px 0;
3    font-family: 'DM Sans', sans-serif;
4    font-size: 16px;
5    color: #333;
6    background-color: #fff;
7  }
8
9 .page-title {
10   margin-bottom: 80px;
11   text-align: center;
12   font-size: 36px;
13   color: #333;
14 }
15
16 .section {
17   margin-bottom: 50px;
18 }
19
20 .section-title {
21   font-size: 24px;
22   text-align: center;
23   margin-bottom: 50px;
24   color: #333;
25 }
26
27 .table {
28   width: 80%;
29   margin-left: auto;
30   margin-right: auto;
31   border-collapse: collapse;
32 }
33
34 .table th,
35 .table td {
36   padding: 12px;
37 }
```

## 5.4 Python Code

### 5.4.1 views.py



The image shows two tabs of Python code in Visual Studio Code:

- views.py** (Top Tab):
 

```

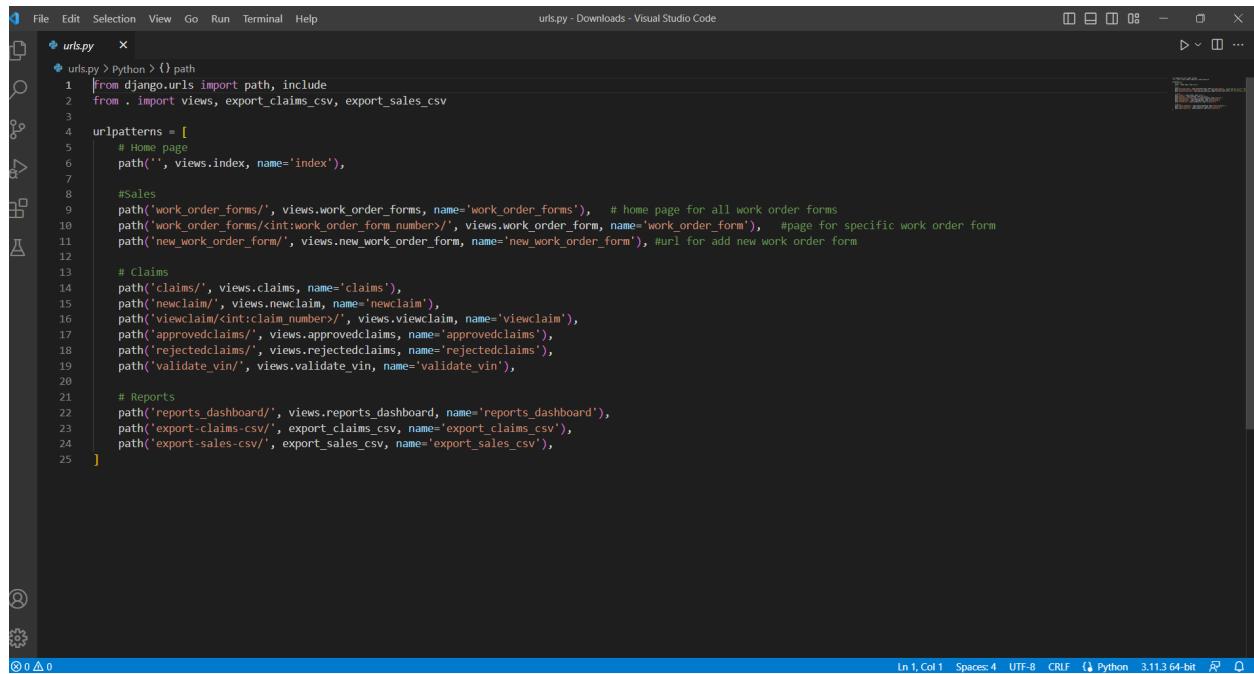
160 ##### REPORTS #####
161 def reports(request):
162     # Process data from claims page and sales page
163     claims_data = Claim.objects.all()
164     sales_data = Sale.objects.all()
165
166     # Pass data to the template
167     context = {
168         'claims_data': claims_data,
169         'sales_data': sales_data,
170     }
171
172     # Render the template
173     return render(request, 'reports.html', context)
174
175
176 def export_sales_csv(request):
177     sales = Sale.objects.all()
178
179     # Create a CSV response
180     response = HttpResponse(content_type='text/csv')
181     response['Content-Disposition'] = 'attachment; filename="sales-report.csv"'
182
183     # Write the CSV data to the response
184     writer = csv.writer(response)
185     writer.writerow(['Dealership', 'Daily Sales', 'Total Revenue'])
186     for sale in sales:
187         writer.writerow([sale.dealership, sale.daily_sales, sale.total_revenue])
188
189     return response
190
191 def export_claims_csv(request):
192     claims = Claim.objects.all()
193
194     # Create a CSV response
195     response = HttpResponse(content_type='text/csv')
196     response['Content-Disposition'] = 'attachment; filename="claims-report.csv"'
197
198     # Write the CSV data to the response
199     writer = csv.writer(response)
200     writer.writerow(['Number of Claims'])
201     writer.writerow([claims.count()])
202
203     return response
      
```
- claims.py** (Bottom Tab):
 

```

109     return response
110
111 def export_claims_csv(request):
112     claims = Claim.objects.all()
113
114     # Create a CSV response
115     response = HttpResponse(content_type='text/csv')
116     response['Content-Disposition'] = 'attachment; filename="claims-report.csv"'
117
118     # Write the CSV data to the response
119     writer = csv.writer(response)
120     writer.writerow(['Number of Claims'])
121     writer.writerow([claims.count()])
122
123     return response
      
```

The status bar at the bottom of the interface shows: In 1, Col 1 | Spaces: 4 | UTF-8 | Python 3.11.3 64-bit |  

## 5.4.2 urls.py



```

File Edit Selection View Go Run Terminal Help
urls.py - Downloads - Visual Studio Code
urls.py > Python > {} path
1  from django.urls import path
2  from . import views, export_claims_csv, export_sales_csv
3
4  urlpatterns = [
5      # Home page
6      path('', views.index, name='index'),
7
8      # Sales
9      path('work_order_forms/', views.work_order_forms, name='work_order_forms'), # home page for all work order forms
10     path('work_order_forms/<int:work_order_form_number>', views.work_order_form, name='work_order_form'), # page for specific work order form
11     path('new_work_order_form/', views.new_work_order_form, name='new_work_order_form'), # url for add new work order form
12
13      # Claims
14      path('claims/', views.claims, name='claims'),
15      path('newclaim/', views.newclaim, name='newclaim'),
16      path('viewclaim/<int:claim_number>', views.viewclaim, name='viewclaim'),
17      path('approvedclaims/', views.approvedclaims, name='approvedclaims'),
18      path('rejectedclaims/', views.rejectedclaims, name='rejectedclaims'),
19      path('validate_vin/', views.validate_vin, name='validate_vin'),
20
21      # Reports
22      path('reports_dashboard/', views.reports_dashboard, name='reports_dashboard'),
23      path('export-claims-csv/', export_claims_csv, name='export_claims_csv'),
24      path('export-sales-csv/', export_sales_csv, name='export_sales_csv'),
25  ]

```

The screenshot shows the Visual Studio Code interface with the file 'urls.py' open. The code defines several URL patterns using the Django path() function. It includes paths for the home page, work order forms (both general and specific by number), claims (including a new claim and viewing individual claims), and reports (a dashboard and CSV exports for claims and sales). The code uses the 'views' module and specific functions like 'export\_claims\_csv' and 'export\_sales\_csv'.