**Django Car Enquiry Form Project Documentation**

**Project Overview**

The Django Car Enquiry Form project is a web application that allows users to submit enquiries about car models, variants, and related information. The collected data is stored in an Excel file for easy management and tracking.

**Project Structure**

**1. Project Name**

* **car\_enquiry\_form**

**2. Application Name**

* **enquiry**

**3. Files and Folders**

* **Project folder**: car\_enquiry\_form
  + settings.py: Configuration for the Django project.
  + urls.py: Routing URLs for the project.
* **App folder**: enquiry
  + views.py: Logic for handling user requests.
  + forms.py: Form definitions.
  + urls.py: Application-specific URLs.
  + templates/: HTML templates for rendering the user interface.
* **Static Folder**: Contains CSS and JavaScript files.
* **Excel File**: car\_enquiry\_data.xlsx stores user data.

**Implementation Details**

**1. Form Fields**

The form collects the following details from users:

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Required** |
| Car Brand | Drop-down | Yes |
| Car Model | Drop-down | Yes |
| Car Year | Drop-down | Yes |
| Car Variant | Drop-down | Yes |
| City | Drop-down | Yes |
| State | Drop-down | Yes |
| First Name | Text input | Yes |
| Last Name | Text input | No |
| Mobile Number | Text input | Yes |
| Email Address | Email input | No |
| Preferred Contact Method | Radio buttons | Yes |
| Enquiry Message | Text area | No |
| Test Drive Request | Checkbox | No |

**2. Backend Configuration**

**Form Definition (forms.py)**

The form uses Django's forms.Form class to define all fields. Required fields are marked, and validation is applied to fields such as mobile\_number to ensure it contains exactly 10 digits.

**Views (views.py)**

* **handle\_car\_enquiry\_form**: Handles form display, submission, and saving data to an Excel file.
* **success\_page**: Displays a success message after form submission.

**Excel Integration**

The openpyxl library is used to:

1. Create an Excel file if it does not exist.
2. Append enquiry data to the Excel sheet upon form submission.

**Routing (urls.py)**

* **Project URLs**:

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('enquiry.urls')),

]

**App URLs**:

from django.urls import path

from . import views

urlpatterns = [

path('', views.handle\_car\_enquiry\_form, name='car\_enquiry\_form'),

path('success/', views.success\_page, name='car\_enquiry\_success'),

]

**3. Templates**

**Form Template (templates/car\_enquiry\_form.html)**

This file contains the HTML structure for the form with CSS for centering and JavaScript for validation and interactivity.

**Success Template (templates/success.html)**

A simple success message with a "Go Back" link to the form.

**Features**

1. **Responsive Form**: Ensures compatibility across devices.
2. **Validation**: Ensures correct data entry using Django validators.
3. **Excel Storage**: Saves all submissions to an Excel file for easy access.
4. **Success Page**: Provides feedback to users after submission.
5. **JavaScript Enhancements**: Adds interactivity and styling.

**Installation and Setup**

**Clone the Repository**:

git clone <repository\_url>

cd car\_enquiry\_form

**Install Dependencies**:

pip install django openpyxl

**Run Migrations**:

python manage.py migrate

**Start the Development Server**:

python manage.py runserver

**Access the Application**:

Open a browser and navigate to http://127.0.0.1:8000/.

**Future Enhancements**

1. **Database Integration**: Use a database like PostgreSQL or SQLite instead of Excel.
2. **Email Notifications**: Send an automated email confirmation to the user.
3. **Admin Panel Improvements**: Add the ability to view submissions in the Django admin panel.
4. **Advanced UI**: Use frameworks like Bootstrap for enhanced styling.

**Conclusion**

This Django Car Enquiry Form project is a simple yet effective way to manage customer enquiries. It demonstrates the use of Django forms, Excel integration, and user-friendly features, making it suitable for real-world applications.