- **django-admin startproject IME**
- **cd IME**
- **python manage.py startapp IME**
- **регистрираш апликација во settings.py**
- **додаваш потребни работи за слики(подолу конкретно напишано)**
- **Креираш модели**
- **python manage.py makemigrations**
- **python manage.py migrate**
- **python manage.py createsuperuser**
- **python manage.py runserver**
- **Проверуваш дали работи /admin**
- **Пишуваш админ**
- **Додаваш нов фајл за форма / пишуваш форма(подолу има и за add, edit, delete)**
- **Додаваш датотека templates, додававаш HTML фајлови во неа**
- **Пишуваш views.py**
- **Регистрираш во urls.py**

SETTINGS.PY

```
import os
MEDIA_ROOT = os.path.join(BASE_DIR, "data/")
MEDIA_URL = '/data/'
```

URLS.PY

```
from django.contrib import admin
from django.urls import path
from django.conf.urls.static import static
from django.conf import settings

from RepairsApp import views

urlpatterns = ([
    path('admin/', admin.site.urls),
    path('index/', views.index, name='index'),
    path('repairs/', views.repairs, name='repairs'),
    path('repair/edit/<id>/', views.edit_repair, name="edit repair"),
    path('repair/delete/<id>/', views.delete_repair, name="delete repair"),
    path('repair_details/<id>/', views.repair_details, name="repair details"),
    ]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT))
```

```
MODELS.PY
from django.db import models
from django.contrib.auth.models import User

class Flight(models.Model):
    code = models.CharField(max_length=100)
    airport_takeoff = models.CharField(max_length=100)
    airport_landing = models.CharField(max_length=100)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    image = models.ImageField(upload_to='data/')
    balloon = models.ForeignKey(Balloon, on_delete=models.CASCADE)
    pilot = models.ForeignKey(Pilot, on_delete=models.CASCADE)
    company = models.ForeignKey(Company, on_delete=models.CASCADE)
```

ADMIN.PY

```python
from django.contrib import admin

from .models import *


class ManufacturerInline(admin.StackedInline):
    model = WorkshopManufacturer
    extra = 1

class RepairAdmin(admin.ModelAdmin):
    exclude = ('user',)
    list_display = ('user',)
    def save_model(self, request, obj, form, change):
        obj.user = request.user
        super().save_model(request, obj, form, change)
class WorkshopAdmin(admin.ModelAdmin):
    inlines = [ManufacturerInline]
    def has_change_permission(self, request, obj=None):
        return obj and obj.user == request.user
    def has_delete_permission(self, request, obj=None):
        return False

class ManufacturerAdmin(admin.ModelAdmin):
    list_display = ('name',)

    def has_add_permission(self, request, obj=None):
        return request.user.is_superuser

class CarAdmin(admin.ModelAdmin):
    list_display = ('type', 'max_speed',)

admin.site.register(Car, CarAdmin)
admin.site.register(Manufacturer, ManufacturerAdmin)
admin.site.register(Workshop, WorkshopAdmin)
admin.site.register(Repair, RepairAdmin)
```

FORMS.PY

```python
from django import forms
from .models import Repair
class RepairForm(forms.ModelForm):
    class Meta:
        model = Repair
        exclude = ['user']
        widgets = {

            'date_time': forms.DateTimeInput(attrs={'type': 'datetime-local'}),
        }

    def __init__(self, *args, **kwargs):
        super(EventForm, self).__init__(*args, **kwargs)
        for visible in self.visible_fields():
            visible.field.widget.attrs['class'] = 'form-control'
            if visible.name == 'is_open':
                visible.field.widget.attrs["class"] = "form-check-input"
```

VIEWS.PY

```
from django.shortcuts import render, redirect
from .forms import RepairForm
from .models import Repair
def index(request):
    return render(request, "index.html")
def repairs(request):
    if request.method == "POST":
        form_data = RepairForm(data=request.POST, files=request.FILES)
        if form_data.is_valid():
            repair = form_data.save(commit=False)
            repair.user = request.user
            repair.image = form_data.cleaned_data['image']
            repair.save()
            return redirect("/repairs")
    qs = Repair.objects.filter(user=request.user, car__type="sedan")
    return render(request, "repairs.html", context={'repairs': qs, 'form': RepairForm})

def edit_repair(request, id):
    repair_instance = Repair.objects.filter(id=id).get()
    if request.method == "POST":
        repair = RepairForm(request.POST, instance=repair_instance)
        if repair.is_valid():
            repair.save()
        return redirect("repairs")
    else:
        repair = RepairForm(instance=repair_instance)
    return render(request, "edit_repair.html", {"form": repair})

def delete_repair(request, id):
    repair_instance = Repair.objects.filter(id=id).get()
    if request.method == "POST":
        repair_instance.delete()
        return redirect("repairs")
    return render(request, "delete_repair.html")

def repair_details(request, id):
    return render(request, "repair_details.html", context={'repair':
Repair.objects.get(id=id)})
```

VIEW SO VNESUVANJE NA OBJEKTI SO ZAPIRKA

```
def event(request):
    if request.method == 'POST':
        form_data = EventForm(data=request.POST, files=request.FILES)
        if form_data.is_valid():
            bands = form_data.cleaned_data['bands']
            band_list = bands.split(',')

            event_object = form_data.save(commit=False)
            event_object.user = request.user
            event_object.image = form_data.cleaned_data['image']
            event_object.num_participants = len(band_list)
            event_object.save()

            for band in band_list:
                band = Band.objects.get(name=band)
                band.events_count = band.events_count + 1
                band.save()
                EventBand.objects.create(band=band, event=event_object)
            return redirect("/events/add")

    return render(request, "events.html", context={'form': EventForm})
```

KARTICKI

```
<div class="row row-cols-md-3 g-4" style="width: 80%; margin: auto">
    {% for repair in repairs%}
  <div class="col">
    <div onclick="window.location.replace('/repair_details/{{ repair.id }}')"
class="card" style="background: rgba(34,34,34,0.83); margin: 10%">
      <img src="{{MEDIA_URL}}{{repair.image.url}}" class="card-img-top" alt="...">
      <div class="card-body">
        <h4 class="card-title" style="color: white">{{repair.code}}</h4>
        <p class="card-text" style="color: gray">{{repair.car.type}}</p>
        <p class="card-text" style="color: white">{{repair.description}}</p>
        <a class="btn" style="border: 3px solid red; color: red">Fix</a>
        <a class="btn btn-primary" href="{% url 'edit repair' repair.id %}">Edit</a>
        <a class="btn btn-primary" href="{% url 'repair details' repair.id
%}">Details</a>
        <a class="btn btn-primary" href="{% url 'delete repair' repair.id %}">Delete</a>
      </div>
    </div>
  </div>
    {% endfor %}
</div>
```

ADD FORM

```
<div>
        <form action="" style="width: 80%;margin: auto" method="post"
enctype="multipart/form-data">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit" class="btn btn-primary">Submit</button>
        </form>
</div>
```

EDIT FORM

```
<form method="post">
        {% csrf_token %}
        {{ form }}
        <input type="submit" value="Submit">
</form>
```

DELETE FORM

```
<h2>Are you sure you want to delete this repair?</h2>
    <form method="post" style="width: 80%; margin: auto;">
        {% csrf_token %}
        <input class="btn btn-danger" type="submit" value="Yes">
        <a class="btn btn-info" href="/repairs">No</a>
</form>
```
