Part 1 (Theory Questions)

Q1. Ans →

i) python -m venv environment and environment\Scripts\activate
ii) pip install django
iii) django-admin startproject college
iv) python –version OR python3 –version

Q2. Ans →

i) python manage.py startapp user
ii) python manage.py makemigrations Then python manage.py migrate
iii) python manage.py runserver
iv) python –version

Q3. Ans →

`render` in Django combines a template with a context, returning an HttpResponse object with that rendered text, mainly used to display HTML pages. `redirect` sends a response to the browser instructing it to go to a different URL, useful for scenarios like post-form submission redirection.

To link static files in an HTML template, you first need to load Django's static file handling mechanism at the top of your template with `{% load static %}`. Then, use the `{% static 'path/to/file' %}` tag to correctly reference static files, such as in a `<link>` tag for CSS files.

Q4.Ans→

HTML →templates folder

CSS files, images, and JavaScript files are saved in a **static** folder, often with subdirectories like **css/**, **img/**, and **js/**.

The **views.py** file handles the business logic by defining views that process user requests, interact with models, and render templates, effectively linking URL patterns with appropriate actions and responses in the application.

Q5.Ans→

The `urls.py` file in Django maps URLs to specific views, defining how web addresses are handled within the application. Models are Python classes in Django that define the structure of database tables, representing data attributes and behaviors of the stored information, essentially facilitating database interactions.

Q6.Ans→

In the Django `settings.py` file, setting "DEBUG = True" enables debugging mode, providing detailed error messages and enabling various debugging tools. If set to "DEBUG = False", Django suppresses error messages and serves static files directly, suitable for production environments but less helpful for debugging as it doesn't display detailed error messages to users.

Part 2 (Practical/coding questions)

Q1.Ans→

i)urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('login/', views.login_user, name='login'),
]
```

ii)views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login
from django.contrib import messages
from django.contrib.auth.forms import AuthenticationForm
def login_user(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:
                login(request, user)
                return redirect('index')  # Redirect to the index page after login
            else:
                messages.error(request, 'Invalid username or password.')
    else:
        form = AuthenticationForm()
    return render(request, 'login.html', {'form': form})
```

iii)login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>
<body>
    <h1>Login</h1>
    {% if messages %}
        {% for message in messages %}
            <p>{{ message }}</p>
        {% endfor %}
    {% endif %}
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" value="Login">
```

```
    </form>
</body>
</html>
```

Q2.Ans→

I)urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('logout/', views.logout_user, name='logout'),
]
```

Ii)views.py

```
from django.shortcuts import redirect
from django.contrib.auth import logout
def logout_user(request):
    logout(request)
    return redirect('index')  # Redirect to the index page after logout
```

iii)index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
</head>
<body>
    <h1>Welcome to our website!</h1>
    <p>This is the home page.</p>
    <a href="{% url 'logout' %}">Logout</a>
</body>
</html>
```

Q3.Ans→

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from django.contrib.auth import authenticate, login
from django.contrib import messages
from django.contrib.auth.forms import AuthenticationForm
@login_required
def index(request):
    return render(request, 'index.html')
def login_user(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
```

```
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:
                login(request, user)
                return redirect('index')
            else:
                messages.error(request, 'Invalid username or password.')
        else:
            form = AuthenticationForm()
        return render(request, 'login.html', {'form': form})
```

Q4.Ans→

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Message Display</title>
</head>
<body>
    {% if messages %}
        <ul class="messages">
            {% for message in messages %}
                <li class="{{ message.tags }}">{{ message }}</li>
            {% endfor %}
        </ul>
    {% endif %}

    <h2>Sample Content</h2>
    <!-- Your other HTML content goes here -->
</body>
</html>
```

Q5. Ans→

```
from django.db import models
class Student(models.Model):
    full_name = models.CharField(max_length=100)
    class_name = models.CharField(max_length=50)
    section = models.CharField(max_length=10)
    roll_number = models.CharField(max_length=20)
    phone_number = models.CharField(max_length=15)
    email = models.EmailField()
    address = models.CharField(max_length=255)
    def __str__(self):
        return self.full_name
```

Q6. Ans→

i) views.py

```
from django.shortcuts import render
```

```
from django.contrib import messages
def index(request):
    messages.success(request, 'Welcome to our website!')
    return render(request, 'index.html')
```

ii) index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome Page</title>
</head>
<body>
    <h1>Welcome to our website!</h1>
    {% if messages %}
        <ul class="messages">
            {% for message in messages %}
                <li>{{ message }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</body>
</html>
```