Test 4: Django Template System

Section A: Multiple Choice

1. What is the purpose of the Django template system?
   - A) To generate Python code dynamically
   - B) To dynamically generate HTML, XML, or other text-based formats
   - C) To manage database connections
   - D) To handle user authentication
2. Which syntax is used for template tags in Django?
   - A) Double curly braces {{ }}
   - B) Single curly braces {% %}
   - C) Square brackets []
   - D) Parentheses ()
3. What can be included in Django templates using template tags?
   - A) Variable outputs and loops
   - B) Function definitions
   - C) Class definitions
   - D) Database queries
4. How are filters applied to variables in Django templates?
   - A) Using double curly braces
   - B) Using single curly braces
   - C) Using parentheses
   - D) Using the pipe character |
5. Where are templates typically stored in a Django project?
   - A) In the root directory
   - B) In the static directory
   - C) In the templates directory of each app
   - D) In the media directory

Section B: Practical Application

6. **HTML Template and Student List**

   a) **Create HTML Template**: Create an HTML template named `student_list.html` that displays a list of students. Each student should be displayed with their first name, last name, and age.

   b) **Retrieve Student Data**: Assume you have a view named `student_list_view` that retrieves a list of `Student` objects from the database and renders the `student_list.html` template. Provide the Django view function syntax to retrieve all `Student` objects and pass them to the `student_list.html` template.

7. **Conditional Logic (if-else)**

   a) **Apply If-Else in Template**: In the `student_list.html` template, implement a conditional statement using Django template tags. Display a message "Adult" next to a student's name if their age is 18 or above; otherwise, display "Minor".

   Ex : -

   {% if student.age == 18 %} Adult {% else %} Minor {% endif %}

Instructions:

- Section B requires you to create an HTML template, retrieve data from Django models, and apply conditional logic using Django template tags.
- For each question, provide clear and concise answers or code snippets demonstrating your understanding.
- Ensure your solutions follow Django's template syntax guidelines.
- Each correctly implemented solution or syntax demonstration will receive 2 points.

Submission:

•Submit your completed question paper along with the PDF document containing screenshots of the HTML pages.

•Submit your PDF document before 10 pm by attaching it to your submission in Whatsapp
Gruop.