**TASK-3**

1. **What are the key features of the Django Template language, and how do they contribute to its advantages in web development?**

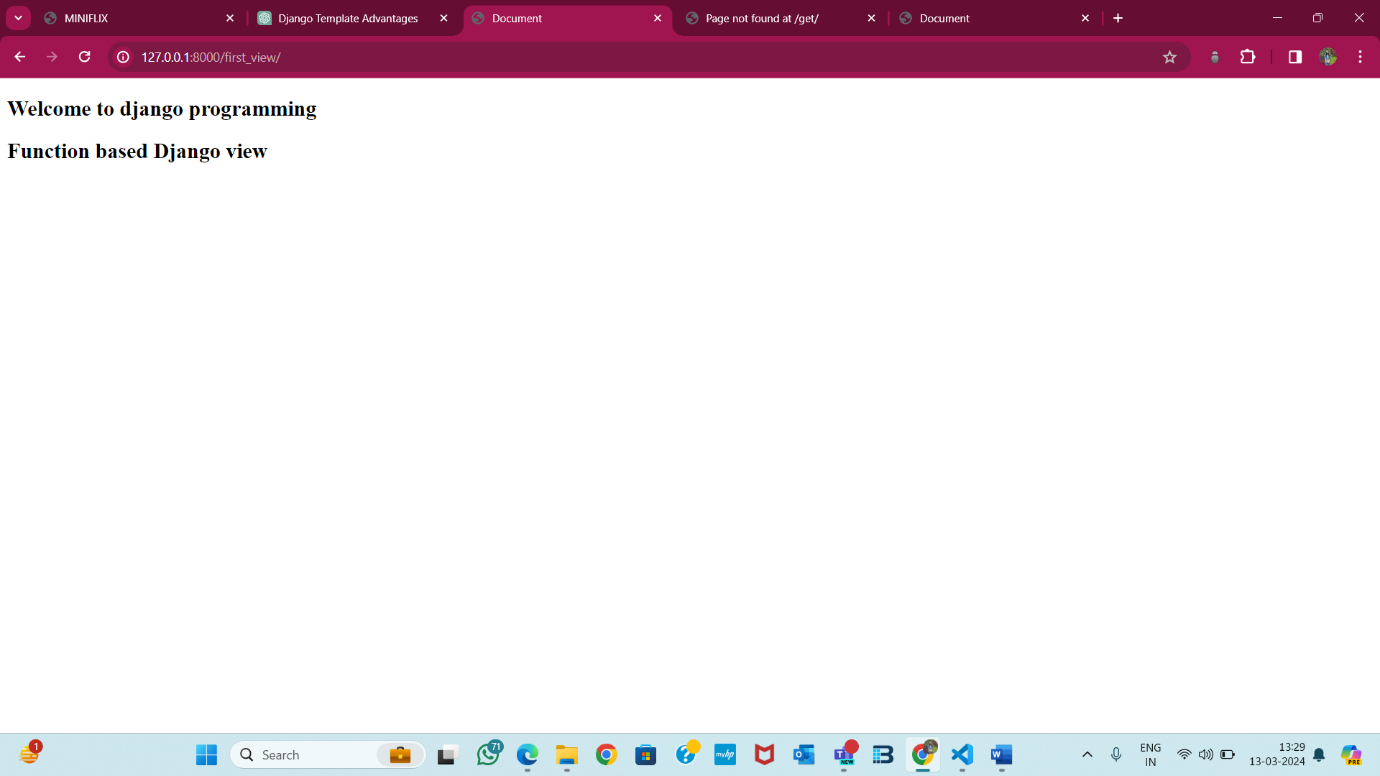
Key features of the Django Template language:

* **Simple Syntax**: Easy-to-read syntax using template tags {% %} and variables {{ }}.
* **Template Inheritance**: Allows for creating base templates with common elements and extending or overriding sections in child templates.
* **Template Filters**: Built-in filters for manipulating and formatting data directly within templates.
* **Template Tags**: Provides tags for logic, iteration, inclusion of other templates, and control flow within templates.
* **Template Context**: Access to a context containing variables and data passed from views to templates.
* **Security**: Automatic HTML escaping by default to prevent cross-site scripting attacks.
* **Integration with Django ORM**: Seamless integration with Django's Object-Relational Mapping system for displaying database objects in templates.
* **Extensibility**: Ability to create custom template tags, filters, and loaders to extend functionality as needed.

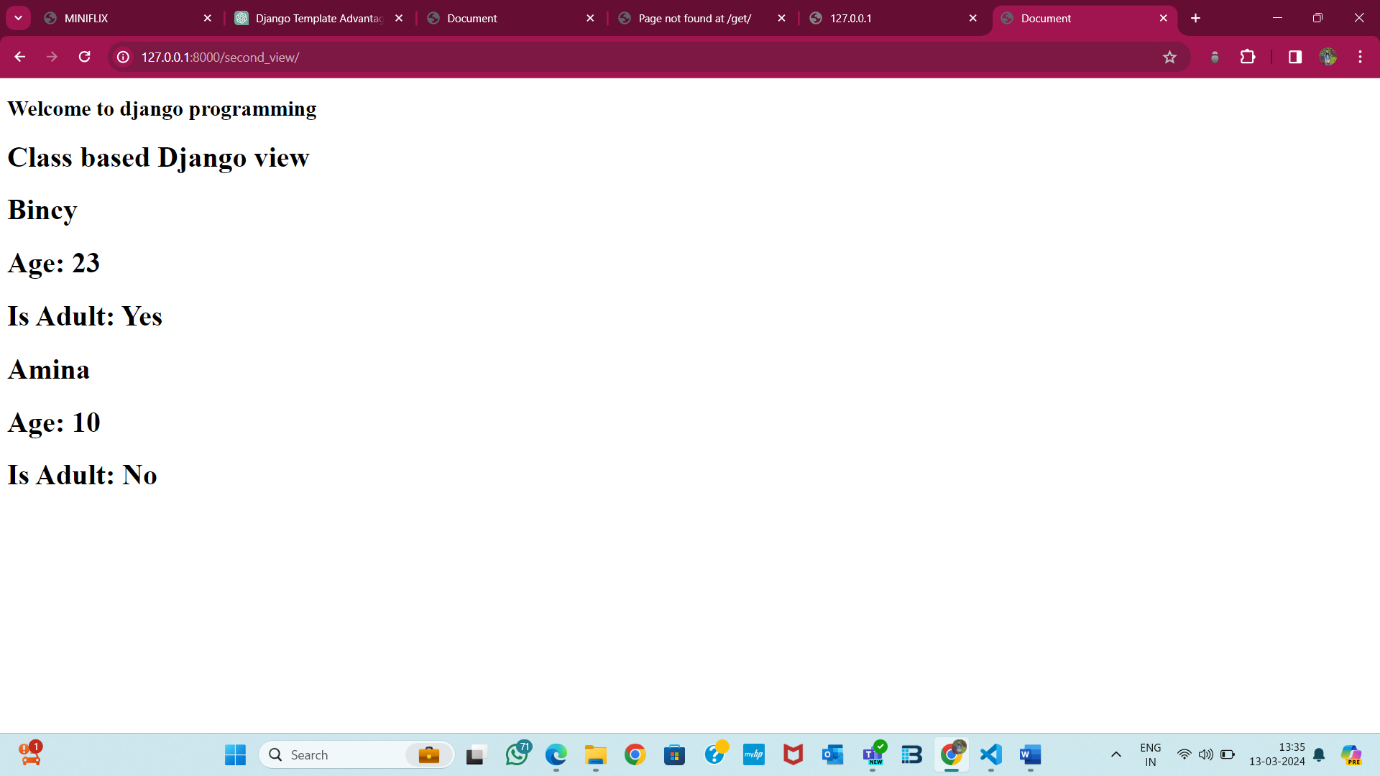
Overall, the Django template language's key features contribute to its advantages in web development by promoting code reusability, readability, security, and flexibility, ultimately enabling developers to build dynamic and feature-rich web applications efficiently.

1. **Create a Django view using both function-based and class-based approaches. Utilize different Django Template Language (DTL) tags within these views.(attach screenshots of SITE)**

**Function Based Django view**

****

**Class Based Django view**

****

**Views.py**

**A screen shot of a computer

Description automatically generated**