1. Preserving State in HTTP Applications

HTTP is a stateless protocol, meaning each request from a client to a server is processed independently, without memory of previous interactions. To preserve the state of an application — especially for user authentication and sessions — web applications use mechanisms like **cookies**, **sessions**, and **tokens**. When a user logs in, the server creates a unique session identifier (session ID) and stores it in a cookie on the client's browser. Each time the client sends a request, that cookie is sent along, allowing the server to identify the user and maintain their logged-in state. In modern web applications, **JSON Web Tokens (JWT)** are also used to handle stateless authentication, storing encoded user data that can be verified without server-side session storage.

2. Performing Django Database Migrations to MariaDB

To migrate a Django project to a server-based relational database like **MariaDB**, you must first install the MariaDB server and Python's `mysqlclient` library, which allows Django to communicate with the database. In the Django project's `settings.py`, the DATABASES section is updated to include the MariaDB engine (`'django.db.backends.mysql'`), along with the host, database name, username, and password. After configuring the connection, the developer runs the commands `python manage.py makemigrations` to prepare migration files and `python manage.py migrate` to apply them to the MariaDB database. This process ensures that all Django models are created as tables in MariaDB, keeping the structure consistent with the application's design.