Research Findings

1. **HTTP State Preservation and Session Management:** HTTP is inherently a stateless protocol, meaning each request from a client to a server is independent and carries no information about previous requests. To preserve the state of an application, mechanisms such as cookies, sessions, and tokens are used. When a user logs in, a server typically creates a session, storing user-specific information on the server-side (e.g., session ID, user details). A cookie containing the session ID is sent to the client and returned with each subsequent request, allowing the server to recognize the user. This approach supports user authentication and maintains state across multiple request-response cycles, enabling functionalities like user-specific dashboards, shopping carts, and personalized content.

2. **Django Database Migrations to a Server-Based Database (MariaDB):** Django manages database schema changes using migrations. To perform migrations on a server-based database like MariaDB, first configure the `DATABASES` setting in `settings.py` with the appropriate engine (`django.db.backends.mysql`), database name, user credentials, host, and port. Install the necessary database driver (e.g., `mysqlclient`). After configuration, run `python manage.py makemigrations` to generate migration files for changes in models, followed by `python manage.py migrate` to apply these migrations to the MariaDB server. This process ensures the server database schema is synchronized with Django models, maintaining data integrity and supporting version-controlled database evolution.