



# Web application programming

## E-PET Project

---

Hassan Jaber - 330214

Instructor: Ostrek Grzegorz

## Project description:

**Overview:** A convenient and safe marketplace to buy and sell pet supplies and toys is what the E-Pet e-commerce platform offers to pet owners and pet product providers. This marketplace offers a vast selection of products and brands and caters to a wide spectrum of pets, including dogs, cats, birds, reptiles, and more. Pet enthusiasts may find everything they need on E-Pet's user-friendly and feature-rich website, which aspires to be the go-to place for animal lovers.

Every user on E-pet can be a buyer or a seller, like for example OLX.

### Key Features:

1. User Registration and Authentication:
  - Users can create accounts, providing personal details.
  - Secure authentication mechanisms, such as email verification and password reset.
2. Product Listings:
  - Registered users can create product listings for pet foods and toys they want to sell.
  - Detailed product descriptions, including images, product specifications, pricing, and seller information.
3. Search and Filters:
  - Users can easily search for specific products.
  - Filter options between foods and toys for refining search results.
4. Shopping Cart:
  - Users can add products to their shopping cart for later purchase.
  - Cart summary and the ability to review, update, or remove items.
5. Checkout and Payment:
  - Secure and user-friendly checkout process.
  - Multiple payment options, including credit/debit cards and digital wallets.
6. User Profiles:
  - User profiles displaying their contact information, order history, and listings.
7. Seller Dashboard:
  - Sellers have access to a dashboard where they can manage their listings from add a product to edit and remove a product, and they can see their product order list, and select each order by clicking on the id number to see the customer information.
8. Security and Privacy:
  - Implement security measures to protect user data and transactions.
  - Ensure privacy compliance, including data protection.
9. Responsive Design:
  - A mobile-responsive website to provide a seamless shopping experience on

various devices.

10. Admin Panel:

- An admin panel to manage and monitor the platform, including user accounts, product listings, and resolving disputes.

11. Restrictions:

- Only signup and logged in users can access their store and their account templates or those buttons wouldn't be displayed in the menu
- Only signup and logged in users can checkout.

12. Forms validation:

- All forms in this applications are validated, like signup, log in, add and edit products and checkout.

13. Automated test:

- For automated test I used selenium framework, Selenium is an open-source framework for automating web browsers. It provides a set of tools and libraries for controlling web browsers through programs and performing various actions, such as navigating web pages, clicking links and buttons, submitting forms, and extracting data. Selenium is widely used for web application testing, web scraping, and automating repetitive tasks that involve interacting with web pages. Selenium is widely used for web application testing, including functional testing, regression testing, and compatibility testing. Testers and developers use Selenium to ensure that web applications work as intended across different browsers and platforms. You can see the code is in core/test.py

**How run project:**

1. Open cmd, and change directory to \EPET\EPET\_env\Scripts and then use command 'activate' to activate the virtual environment
2. Change directory to \EPET\EPET then use command 'python manage.py runserver'
3. Go to web browser and surf to the localhost

**Database:** To access the database browse localhost/admin, and the credential are username:admin, and password: 0000

**Dataflow:**

Data flows in my application through a specific pattern based on the Model-View-Controller (MVC) architectural design pattern that Django follows. In Django, this pattern is known as Model-View-Template (MVT).

1. URL Routing:

- When a user makes a request to a Django web application, the request is first received by Django's URL dispatcher.
- The URL dispatcher maps the URL to a specific view function using URL patterns defined in the application's URLs configuration.

2. View (Controller):

- The view function is responsible for processing the incoming HTTP request.
- It may involve data retrieval, processing, and rendering of a response.
- Views can interact with models to fetch or update data.

### 3. Model (Model):

- The model represents the data structure and the database schema for the application.
- Models define the data structure and how data should be stored and retrieved from the database.
- Django's Object-Relational Mapping (ORM) provides an abstraction for working with the database.

### 4. Database Interaction:

- When a view requires data from the database, it interacts with the model layer to fetch or update data.
- 

### 5. Template (View):

- Once the view has processed the request and obtained data, it typically renders an HTML template.
- The template is a separate file that contains the structure and layout of the web page. It may include placeholders for dynamic data.

### 6. Rendered Response (View):

- The template, combined with the data obtained in the view, generates an HTML response that is sent back to the user's browser.

### 7. HTTP Response:

- The Django view sends an HTTP response to the user's browser, which contains the HTML content of the web page.

### 8. User Interaction (Client-Side):

- The user interacts with the rendered web page in their browser by clicking links, submitting forms, or making additional requests.

### 9. URL Routing (Client-Side):

- When the user interacts with the web page, further requests are sent to the server using URLs, and the process repeats.

## **Screenshots of my application:**



pettoy3  
\$15.15



pettoy2  
\$55.55



petToy1  
\$14.15



petfood3  
\$12.34



PetFood2  
\$14.14



PetFood1  
\$12.34

My Account

User Information

**Username:** admin  
**First Name:** Hassan

**Email:** admin@petnet.com  
**Last Name:** Jaber

Order History

You don't have any orders yet.

Log Out

My Store

My Orders

Order ID	Title	Quantity	Price
1	pettoy3	1	\$15.15

Add Product

My Products



pettoy3  
\$15.15

Status: Active



pettoy2  
\$55.55

Status: Active



petToy1  
\$14.15

Status: Active

Order Detail - 1

Product Details

Title	Quantity	Price
pettoy3	1	\$15.15

Customer Information for This Order

**First Name:** Hassan  
**Last Name:** Jaber  
**Address:** Cypryjska 26  
**Zip Code:** 02-761  
**City:** Warsaw

Add product

Category:

.....

Title:

Description:

Price:

Image:

Choose File No file chosen

Submit

## Edit product

Category:

Toys

Title:

pettoy2

Description:

this is a awesome pet toy

Price:

\$555

Image:

Choose File

No file chosen

Submit

Delete

## Login

Username

Password

Login

Don't have an account? [Signup](#)  
[Forgot Password?](#)

## Sign up

Username

Email

First Name

Last Name

Password

Confirm Password

Sign up

Checkout

Total Cost: \$15.15

First name:

Last name:


Address:

Zipcode:

City:

Checkout

Cart



pettoy3  
\$15.15

+

1

-

Remove

Total cost: \$15.15

Checkout

Password Reset

Email Address

Reset Password