Django Web Framework

BASIC THEORY

What is a Web Application?

It is a software program that you access through your web browser, like Chrome or Safari. Unlike software you download and install on your device, web apps run on a remote server and the content is delivered to your device through the internet. This means you can use web apps from any device with a web browser and an internet connection. Examples of web applications include Gmail, Google Docs, and many social media platforms.

Every web application contain 2 main components

1. Frontend
2. Backend

Front-End (Client-side):

- Focuses on the visual design and user experience (UX) of the web application.

- Technologies used: HTML, CSS, JavaScript (and frameworks like React or Angular).

- Skills required: Creativity, understanding of UI/UX principles, strong coding skills.

Back-End (Server-side):

- Handles data storage, processing, and server logic.

- Ensures communication between the database and the front-end.

- Technologies used: Programming languages like Python, Java, or Ruby; databases like MySQL or PostgreSQL.

- Skills required: Strong problem-solving skills, knowledge of databases and server administration.

Static vs Dynamic Websites-

Static website – same content for each and every person

Example- Login Page of Google or any app

Home page before login of any website

Dynamic website- different content for each and every person

Example- Social media, E-commerce, G-mail inbox

Here server-side logic is there, data differs person to person.

Django Web Framework-

Framework- A framework provides faster development as it contains pre-written code for the task that are commonly used in every web applications.

Django is a free and open-source web framework. It is written in python.

It follows the model-template-view (MTV) architectural pattern, which separates the different aspects of web development for better organization and maintainability.

- **Model-Template-View (MTV) Pattern:**

  o Models: Define the data structure of your application, representing data like users, products, or blog posts.

  o Templates: Determine how data is displayed on the user interface (UI). Django uses a templating language like HTML to structure and format content.

  o Views: Handle user requests and interact with models and templates to generate the appropriate response.

Django has a rich set of built-in features, aiming to provide everything you need for most web development tasks "out of the box." This includes functionalities for:

- User authentication and authorization
- Database access and management
- URL routing
- Web forms
- Security features
- Administrative interface

When to Use Django:

- Django is a great choice for developing various web applications, including:
    - Content management systems (CMS)
    - E-commerce platforms
    - Social networking applications
    - News and media websites
    - Custom web applications requiring complex functionalities

Now for Backend why Python and Django is best option?

- **Rapid Development:** Django provides a high-level framework with pre-built features for common web development tasks like user authentication, database interaction, and URL routing. This allows developers to build applications faster and with less boilerplate code.

- **Security:** Django prioritizes security with built-in protection against common web vulnerabilities like SQL injection and cross-site scripting.

- **Scalability:** Django can handle small to large-scale web applications. Its modular design allows for easy expansion as your application grows.

- **Large Community:** Python and Django have vast and active communities, providing ample resources, tutorials, and support for developers.

Some companies that are using Django for their web services-

Here are some examples:

- Social Media: Instagram, Pinterest, Disqus (commenting platform)

- Streaming Services: Spotify

- Entertainment: Netflix (partially)

- News and Media: The Washington Post

- E-commerce: Dropbox

- Others: Mozilla, YouTube (originally), Eventbrite, National Geographic

Join our growing community of tech enthusiasts! Subscribe to our YouTube channel, where we simplify programming languages in the easiest way possible. Gain a deeper understanding with our clear explanations and receive exclusive notes to enhance your learning journey. Don't miss out on valuable insights – hit that subscribe button now and embark on a programming adventure with us!

Subscribe to our channel: https://www.youtube.com/@HTEASYLEARNING