MODULE-1

- Explanation of what a web framework is and its importance in web development.
- Overview of the MVC (Model-View-Controller) design pattern.
- Explanation of the responsibilities of each component in MVC.
- Historical overview of Django's development and evolution.
- Explanation of the typical structure of a Django project.
- Description of directories and files commonly found in a Django project.
- Detailed explanation of Django's architecture.
- Discussion of how Django handles requests and responses.
- Overview of the role of Django's core components such as models, views, and templates.
- Further exploration of Django's features and capabilities.
- Discussion of Django's built-in functionalities for handling authentication, routing, forms, etc.
- Explanation of function-based views (FBVs) and class-based views (CBVs).
- Details on rendering templates and handling form submissions.
- Explanation of how URLs are mapped to view functions or classes in Django.
- Discussion of URL patterns and their configuration in Django URLconfs.
- Detailed explanation of URLconfs in Django.
- Discussion on the concept of loose coupling between URLs and views in Django URL routing.
- Explanation of common errors that can occur in Django URL patterns.
- Discussion on how to handle wild card patterns and resolve errors effectively.

MODULE-2

- What is the purpose of Django's template system?
- How does Django's template system help in separating presentation logic from business logic?
- Explain the syntax of Django templates.
- What are template tags and how are they used in Django templates?
- What are template filters and how are they applied in Django templates?
- How do you integrate Django templates into your Django project?
- Explain the role of the render() function in Django views.
- How can you pass data from views to templates in Django?
- Describe the process of rendering a template in a Django view.
- What are the advantages of using Django's template system compared to raw HTML?
- What are template tags and why are they used in Django templates?
- Provide examples of conditional rendering using template tags.
- How do you loop over data using template tags in Django?
- Explain the purpose of the {% block %} tag in Django templates.
- What are template filters and how do they modify template variables?
- What is template inheritance in Django and why is it useful?
- How do you create a base template in Django?
- Explain the concept of blocks in Django templates.
- How can you extend a base template to create child templates?
- Provide examples of overriding blocks in child templates.

MODULE-3

- How do you activate the Django admin interface for your project?
- What are the benefits of using the Django admin interface?
- How do you access the Django admin interface?
- What actions can you perform using the Django admin interface?
- What are the different ways to customize the Django admin interface?
- Can you provide examples of customizations you can make to the admin interface?
- What is form processing in Django?
- Explain the steps involved in processing a form in Django.
- How do you create a feedback form in Django?
- What are some common features of feedback forms?
- How does Django handle form submissions?
- Explain the process of handling form submissions in Django views.
- What is custom validation in Django forms?
- How do you implement custom validation for form fields?
- What are model forms in Django?
- How do you create a model form for a Django model?
- What is URLconf in Django?
- Can you provide some tips or tricks for working with URL patterns in Django?
- How do you include URL patterns from other apps in Django?
- Why would you use URL namespaces when including other URLconfs?

MODULE-4

- What are generic views in Django?
- Explain the advantages of using generic views over writing views from scratch.
- List and briefly explain some of the generic view classes provided by Django.
- How do you customize generic views in Django?
- Can you use mixins with generic views? Explain with an example.
- Explain the role of the built-in authentication system in Django.
- How do you create and manage user accounts in Django?
- Discuss the different authentication mechanisms supported by Django.
- Can you explain the concept of authorization and how it relates to authentication in web applications?

- What is meant by extending generic views in Django?
- How can you extend generic views in Django? Provide examples.
- Explain the concept of view inheritance in Django.
- What are some common use cases for extending generic views?
- How do you handle data processing and rendering in extended generic views?
- What are the benefits of extending generic views?
- Explain how you can add additional functionality to generic views by extending them.
- Provide examples of scenarios where you might need to extend generic views.
- How do you handle form processing and validation in extended generic views?
- Can you explain the concept of view composition in Django?
- What are MIME types and why are they important in web development?
- Explain the role of MIME types in HTTP headers.
- List and briefly describe some common MIME types used on the web.
- How do you set MIME types in Django views?
- Discuss the security implications of handling MIME types in web applications.
- Why would you need to generate non-HTML content like CSV and PDF in a web application?
- What are the advantages of generating CSV and PDF files dynamically?
- Explain the process of generating CSV files in Django.
- How do you generate PDF files in Django? Mention any libraries you can use.
- Discuss the considerations for handling large datasets when generating CSV and PDF files.
- What is the syndication feed framework in Django?
- Explain the purpose of syndication feeds in web development.
- How do you create and configure syndication feeds in Django?
- Discuss the different types of syndication feeds supported by Django.
- Can you provide examples of how syndication feeds are used in real-world applications?
- What is the sitemap framework in Django?
- Explain the importance of sitemaps for search engine optimization (SEO).
- How do you create and configure sitemaps in Django?
- Discuss the benefits of using the sitemap framework for large websites.
- Can you explain how sitemaps are generated and served to search engines?
- What are cookies and sessions in web development?
- Explain the differences between cookies and sessions.
- How are cookies and sessions managed in Django?
- Discuss the security considerations when using cookies and sessions.
- Provide examples of scenarios where cookies and sessions are used in web applications.
- How does user authentication work in Django?

MODULE-5

- What is AJAX, and how does it enhance the user experience in web applications?
- How can you integrate AJAX into Django projects?
- Explain the difference between synchronous and asynchronous JavaScript.
- What are the different types of HTTP requests supported by AJAX, and how are they used?
- How do you handle AJAX requests and responses in Django views?
- What is the purpose of using JavaScript frameworks like jQuery in Django projects?
- Explain the concept of event handling in JavaScript and how it applies to AJAX interactions.
- How do you serialize Django model data to JSON format for AJAX responses?
- What are the advantages and disadvantages of using XML compared to JSON in AJAX requests?
- How can you implement server-side validation for AJAX requests in Django?
- Describe the process of implementing real-time updates using AJAX in Django.
- How do you prevent cross-site request forgery (CSRF) attacks when using AJAX in Django?
- Explain the role of jQuery plugins in enhancing AJAX functionality in Django applications.
- What are the best practices for optimizing performance when using AJAX in Django?
- How do you troubleshoot common issues encountered when working with jQuery and AJAX in Django projects?