

## VI Semester

FULLSTACK DEVELOPMENT			
Course Code	21CS62	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 20 P	Total Marks	100
Credits	04	Exam Hours	03
<b>Course Learning Objectives:</b> CLO 1.Explain the use of learning full stack web development. CLO 2.Make use of rapid application development in the design of responsive web pages. CLO 3.Illustrate Models, Views and Templates with their connectivity in Django for full stack web development. CLO 4.Demonstrate the use of state management and admin interfaces automation in Django. CLO 5.Design and implement Django apps containing dynamic pages with SQL databases.			
<b>Teaching-Learning Process (General Instructions)</b>  These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.  <ol style="list-style-type: none"><li>1. Lecturer method (L) does not mean only traditional lecture method, but different type of teaching methods may be adopted to develop the outcomes.</li><li>2. Show Video/animation films to explain functioning of various concepts.</li><li>3. Encourage collaborative (Group Learning) Learning in the class.</li><li>4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.</li><li>5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.</li><li>6. Topics will be introduced in a multiple representation.</li><li>7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.</li><li>8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.</li></ol>			
<b>Module-1: MVC based Web Designing</b>  Web framework, MVC Design Pattern, Django Evolution, Views, Mapping URL to Views, Working of Django URL Confs and Loose Coupling, Errors in Django, Wild Card patterns in URLs.			
<b>Textbook 1: Chapter 1 and Chapter 3</b>			
<b>Laboratory Component:</b>  <ol style="list-style-type: none"><li>1. Installation of Python, Django and Visual Studio code editors can be demonstrated.</li><li>2. Creation of virtual environment, Django project and App should be demonstrated</li><li>3. Develop a Django app that displays current date and time in server</li><li>4. Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server.</li></ol>			
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"><li>1. Demonstration using Visual Studio Code</li><li>2. PPT/Prezi Presentation for Architecture and Design Patterns</li><li>3. Live coding of all concepts with simple examples</li></ol>		
<b>Module-2: Django Templates and Models</b>  Template System Basics, Using Django Template System, Basic Template Tags and Filters, MVT Development Pattern, Template Loading, Template Inheritance, MVT Development Pattern.			

Configuring Databases, Defining and Implementing Models, Basic Data Access, Adding Model String Representations, Inserting/Updating data, Selecting and deleting objects, Schema Evolution

**Textbook 1: Chapter 4 and Chapter 5**

**Laboratory Component:**

1. Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event
2. Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages: contact us, About Us and Home page of any website.
3. Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.

**Teaching-Learning Process**

1. Demonstration using Visual Studio Code
2. PPT/Prezi Presentation for Architecture and Design Patterns
3. Live coding of all concepts with simple examples
4. Case Study: Apply concepts learnt for an Online Ticket Booking System

**Module-3: Django Admin Interfaces and Model Forms**

Activating Admin Interfaces, Using Admin Interfaces, Customizing Admin Interfaces, Reasons to use Admin Interfaces.

Form Processing, Creating Feedback forms, Form submissions, custom validation, creating Model Forms, URLConf Ticks, Including Other URLConfs.

**Textbook 1: Chapters 6, 7 and 8**

**Laboratory Component:**

1. For student and course models created in Lab experiment for Module2, register admin interfaces, perform migrations and illustrate data entry through admin forms.
2. Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project.

**Teaching-Learning Process**

1. Demonstration using Visual Studio Code
2. PPT/Prezi Presentation for Architecture and Design Patterns
3. Live coding of all concepts with simple examples

**Module-4: Generic Views and Django State Persistence**

Using Generic Views, Generic Views of Objects, Extending Generic Views of objects, Extending Generic Views.

MIME Types, Generating Non-HTML contents like CSV and PDF, Syndication Feed Framework, Sitemap framework, Cookies, Sessions, Users and Authentication.

**Textbook 1: Chapters 9, 11 and 12**

**Laboratory Component:**

1. For students enrolment developed in Module 2, create a generic class view which displays list of students and detailview that displays student details for any selected student in the list.
2. Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.

**Teaching-Learning Process**

1. Demonstration using Visual Studio Code
2. PPT/Prezi Presentation for Architecture and Design Patterns

	<ol style="list-style-type: none"> <li>Live coding of all concepts with simple examples</li> <li>Project Work: Implement all concepts learnt for Student Admission Management.</li> </ol>
<b>Module-5: jQuery and AJAX Integration in Django</b>	
Ajax Solution, Java Script, XMLHttpRequest and Response, HTML, CSS, JSON, iFrames, Settings of Java Script in Django, jQuery and Basic AJAX, jQuery AJAX Facilities, Using jQuery UI Autocomplete in Django	
<b>Textbook 2: Chapters 1, 2 and 7.</b>	
<b>Laboratory Component:</b> <ol style="list-style-type: none"> <li>Develop a registration page for student enrolment as done in Module 2 but without page refresh using AJAX.</li> <li>Develop a search application in Django using AJAX that displays courses enrolled by a student being searched.</li> </ol>	
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"> <li>Demonstration using Visual Studio Code</li> <li>PPT/Prezi Presentation for Architecture and Design Patterns</li> <li>Live coding of all concepts with simple examples</li> <li>Case Study: Apply the use of AJAX and jQuery for development of EMI calculator.</li> </ol>
<b>Course outcome (Course Skill Set)</b> At the end of the course the student will be able to: CO 1. Understand the working of MVT based full stack web development with Django. CO 2. Designing of Models and Forms for rapid development of web pages. CO 3. Analyze the role of Template Inheritance and Generic views for developing full stack web applications. CO 4. Apply the Django framework libraries to render nonHTML contents like CSV and PDF. CO 5. Perform jQuery based AJAX integration to Django Apps to build responsive full stack web applications,	
<b>Assessment Details (both CIE and SEE)</b>  The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together  <b>Continuous Internal Evaluation:</b>  Three Unit Tests each of <b>20 Marks (duration 01 hour)</b> <ol style="list-style-type: none"> <li>First test at the end of 5<sup>th</sup> week of the semester</li> <li>Second test at the end of the 10<sup>th</sup> week of the semester</li> <li>Third test at the end of the 15<sup>th</sup> week of the semester</li> </ol> Two assignments each of <b>10 Marks</b> <ol style="list-style-type: none"> <li>First assignment at the end of 4<sup>th</sup> week of the semester</li> <li>Second assignment at the end of 9<sup>th</sup> week of the semester</li> </ol>	

Practical Sessions need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **20 marks**.

- Rubrics for each Experiment taken average for all Lab components – 15 Marks.
- Viva-Voce– 5 Marks (more emphasized on demonstration topics)

The sum of three tests, two assignments, and practical sessions will be out of 100 marks and will be **scaled down to 50 marks**

(to have a less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

**CIE methods /question paper has to be designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.**

#### **Semester End Examination:**

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the subject (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks. Marks scored shall be proportionally reduced to 50 marks
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module

#### **Suggested Learning Resources:**

##### **Textbooks**

1. Adrian Holovaty, Jacob Kaplan Moss, The Definitive Guide to Django: Web Development Done Right, Second Edition, Springer-Verlag Berlin and Heidelberg GmbH & Co. KG Publishers, 2009
2. Jonathan Hayward, Django Java Script Integration: AJAX and jQuery, First Edition, Pack Publishing, 2011

##### **Reference Books**

1. Aidas Bendroraitis, Jake Kronika, Django 3 Web Development Cookbook, Fourth Edition, Packt Publishing, 2020
2. William Vincent, Django for Beginners: Build websites with Python and Django, First Edition, Amazon Digital Services, 2018
3. Antonio Mele, Django3 by Example, 3<sup>rd</sup> Edition, Pack Publishers, 2020
4. Arun Ravindran, Django Design Patterns and Best Practices, 2<sup>nd</sup> Edition, Pack Publishers, 2020.
5. Julia Elman, Mark Lavin, Light weight Django, David A. Bell, 1<sup>st</sup> Edition, Oreily Publications, 2014

##### **Weblinks and Video Lectures (e-Resources):**

1. MVT architecture with Django: <https://freevideolectures.com/course/3700/django-tutorials>
2. Using Python in Django: <https://www.youtube.com/watch?v=2BqoLiMT3Ao>
3. Model Forms with Django: <https://www.youtube.com/watch?v=gMM1rtTwKxE>
4. Real time Interactions in Django: <https://www.youtube.com/watch?v=3gHmfoeZ45k>
5. AJAX with Django for beginners: <https://www.youtube.com/watch?v=3VaKNyjlxAU>

##### **Activity Based Learning (Suggested Activities in Class)/ Practical Based learning**

1. Real world problem solving - applying the Django framework concepts and its integration with AJAX to develop any shopping website with admin and user dashboards.

***Short Preamble on Full Stack Web Development:***

*Website development is a way to make people aware of the services and/or products they are offering, understand why the products are relevant and even necessary for them to buy or use, and highlight the striking qualities that set it apart from competitors. Other than commercial reasons, a website is also needed for quick and dynamic information delivery for any domain. Development of a well-designed, informative, responsive and dynamic website is need of the hour from any computer science and related engineering graduates. Hence, they need to be augmented with skills to use technology and framework which can help them to develop elegant websites. Full Stack developers are in need by many companies, who knows and can develop all pieces of web application (Front End, Back End and business logic). MVT based development with Django is the cutting-edge framework for Full Stack Web Development. Python has become an easier language to use for many applications. Django based framework in Python helps a web developer to utilize framework and develop rapidly responsive and secure web applications.*