

Front-End Development

COURSE DESCRIPTION

This course is designed to introduce students to modern Web Development. Especially, for client side - Angular and for server side - Django frameworks.

Angular is a platform and framework for building client applications in HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience. This course gives a complete understanding of Django.

This course is designed for developers who want to learn how to develop quality web applications using the smart techniques and tools offered by Angular and Django. Besides this, students will learn how to solve real world problems from industry.

COURSE OBJECTIVES

The objective of this course is to provide the student with real world tasks from industry and find the best solution for them and work in a team.

COURSE OUTCOMES

In the end of the current course students will know:

- HTML(5), CSS(3), JavaScript
- Node Package Manager (npm)
- Angular Modules, Components, Services, Interfaces
- JavaScript, TypeScript
- Have an intermediate skill level of Python programming.
- Web application architecture, how web works
- Understand steps of web app development
- Build websites using Django 2
- How to create a local development server from scratch
- How to build your own browsable, self documenting REST API
- Working with Django Templates

COURSE POST REQUISITES

Knowledge and skills obtained during study of course Web Development are used in following courses: Programming Technologies, Object-Oriented Programming, Foundation of Web development.

LITERATURE

1. <https://angular.io/>
2. Adam Freeman London, UK ISBN-13 (pbk): 978-1-4842-3648-2
3. The Django Book - MIT 2015
 - a. <http://gs1.mit.edu/media/programs/south-africa-summer-2015/materials/djangobook.pdf>
4. Adrian Holovaty, Jacob Kaplan-Moss, et al
 - a. <https://media.readthedocs.org/pdf/djangobook/latest/djangobook.pdf>
5. Tutorials
 - a. <https://tutorial.djangogirls.org/en/>
 - b. <https://djangoforbeginners.com>
 - c. <https://docs.djangoproject.com/en/2.1/intro>

Week	Class work		Laboratory works
	Topic	Lecture	
1	Introduction to Web Development:	1	<i>1. Laboratory work #1</i>

	<ul style="list-style-type: none"> • What is the website? • How does the Web work? • Technologies in both client and server side • Framework & Library • Back-End framework comparison • Basic techniques for scaling • What is the API? 		
2	Web development roadmap <ul style="list-style-type: none"> • Web development roadmap • HTML Elements • Element attributes • HTML Forms • HTML Forms Inputs • CSS • HTML5/CSS3 	2	1. Laboratory work #2
3	JavaScript <ul style="list-style-type: none"> • JavaScript Standards • Data Types • Variable scoping • Functional Programming • JSON • DOM • Event handling • HTML Element manipulating 	3	1. Laboratory work #3
4	Introduction to Angular. <ul style="list-style-type: none"> • What is the Goal of Angular? • Angular CLI • JavaScript & Typescript 	4	1. Laboratory work #4
5	Angular Components <ul style="list-style-type: none"> • Properties • Data Binding • Templates • Styles • Life-cycle hooks 	5	1. Laboratory work #5
6	Modules, Router Module Getting Data From RESTful APIs <ul style="list-style-type: none"> • Reactive Programming • Services • Observables 	6	1. Laboratory work #6
7	Midterm		
8	Midterm		
9	Introduction to Python PL, Django: <ul style="list-style-type: none"> • Python programming language • What is Django? • Django project structure • Django configurations file (settings.py) • Django router file (urls.py) • Django Web Server Gateway Interface (wsgi.py) 	9	1. Laboratory work #7 2. Project
10	Building REST APIs With Django REST	10	1. Laboratory work #8

	Framework: <ul style="list-style-type: none"> Fundamentals of Basic REST API Design REST API Architecture <ul style="list-style-type: none"> Grouping API URLs Version Your API 		2. <i>Project</i>
11	Generic Views, Sessions, Users, and Registration <ul style="list-style-type: none"> Using Generic Views Generic Views of Objects Django's Session Framework Users and Authentication 	11	1. <i>Laboratory work #9</i> 2. <i>Project</i>
12	DRF Serialization <ul style="list-style-type: none"> Creating a Serializer class Working with Serializers Types of Serializer classes Simple Serializer class ModelSerializers Writing regular Django views using our Serializer 	12	1. <i>Laboratory work #10</i> 2. <i>Project</i>
13	DRF Requests and Responses: <ul style="list-style-type: none"> Request objects Response objects Status codes Wrapping API views Pulling it all together Authentication: <ul style="list-style-type: none"> Adding endpoints for our User models Adding required permissions to views Adding login to the browsable API Authenticating with the API 	13	1. <i>Project defense (front part)</i>
14	Interacting with a Database: Models, The Django Administration Site: <ul style="list-style-type: none"> The MTV Development Pattern Configuring the Database Defining Models in Python Inserting and Updating Data Selecting Objects <ul style="list-style-type: none"> Filtering Ordering Slicing Deleting Objects Making Changes to a Database Schema Activating the Admin Interface 	14	1. <i>Project defense</i>
15	<i>Project defense</i>		1. <i>Project defense 100%</i>