KBTU KAZAKH-BRITISH TECHNICAL UNIVERSITY

JSC «Kazakh-British Technical University»
School of IT and Engineering

APPROVED BY
Dean of SITE

____
«____»__________ 20__.

SYLLABUS

Discipline: Web Development
Number of credits: 4
Term: Spring 20__
Instructors: Bobur Mukhsimbayev, Aibek Kuralbayev

Personal Information about the Instructor	Time and place of classes		Contact information
	Classes	Office Hours	e-mail
Bobur Mukhsimbayev	According to the schedule	Room 184, will be appointed	b.mukhsimbaev@kbtu.kz
Aibek Kuralbayev	According to the schedule	Room 184, will be appointed	a.kuralbaev@kbtu.kz

COURSE DURATION: 4 credits, 15 weeks

COURSE DESCRIPTION

This course provides a practical introduction to modern web development using Angular for the client side and Django for the server side. Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design.

Through hands-on projects and lab assignments, students will gain experience building full-stack web applications. The course focuses on the practical use of Angular and Django in solving real-world problems, with an emphasis on code quality, scalability, and industry best practices.

COURSE OBJECTIVES

- Equip students with skills to design and implement full-stack web applications using Angular and Django.
- Provide experience in solving industry-inspired problems using modern frameworks and tools.
- Foster collaboration and teamwork in software development through project-based learning.
- Reinforce foundational knowledge of web technologies including HTML, CSS, JavaScript, and RESTful API development.

COURSE OUTCOMES

Upon successful completion of the course, students will be able to:

- Demonstrate proficiency in HTML5, CSS3, and JavaScript fundamentals.
- Work with Node Package Manager (npm) to manage Angular dependencies.
- Build and structure Angular applications using components, modules, services, interfaces, and routing.
- Understand and use both JavaScript and TypeScript effectively.
- Apply intermediate-level Python programming skills in web development.
- Understand web architecture and the HTTP request/response lifecycle.
- Develop and deploy Django applications using models, views, templates, and REST APIs.
- Set up a local development environment including virtual environments and development servers.
- Use Django REST Framework to build secure, self-documenting RESTful APIs with serialization and authentication.
- Integrate Angular front-end applications with Django back-end APIs.
- Apply industry best practices for web application development and testing.

COURSE POST REQUISITES

This course provides foundational skills for several development tracks:

- **Frontend Track** – Advanced Angular, UI design, and state management
- **Backend Track** – Django, REST APIs, databases, and deployment
- **Mobile Development Track** – API integration with Flutter or React Native
- **UX/UI Track** – Structure and interaction design for web interfaces
- **Full-Stack Pathway** – Combined client-side and server-side development

LITERATURE

1. *You Don't Know JS (2nd Edition)* – A deep dive into JavaScript concepts:
   - https://github.com/getify/You-Dont-Know-JS/blob/2nd-ed/README.md
2. *Eloquent JavaScript* – A modern introduction to JavaScript:
   - https://eloquentjavascript.net/
3. *Developer Roadmap* – Guides and visual paths for developers:
   - https://github.com/kamranahmedse/developer-roadmap
4. *W3Schools – HTML Tutorial* – Reference for HTML elements and usage:
   - https://www.w3schools.com/html/
5. *Airbnb CSS Style Guide* – Best practices and standards for writing CSS:
   - https://github.com/airbnb/css
6. *Angular Official Documentation* – Learn and build with Angular:
   - https://angular.dev/
7. *PEP 8 – Python Style Guide* – Conventions for Python code:
   - https://peps.python.org/pep-0008/
8. *LearnPython.org* – Interactive Python tutorials:
   - https://www.learnpython.org/
9. *Django Documentation* – Official Django framework docs:
   - https://docs.djangoproject.com/
10. *Django REST Framework* – Build REST APIs with Django:
    - https://www.django-rest-framework.org/
11. *Django Girls Tutorial* – Beginner-friendly Django project tutorial:
    - https://tutorial.djangogirls.org/en/

Week	Classwork		Laboratory works
	Topic	Lecture	
1	**Introduction to Web Development:** ● What is the website? ● How does the Web work? ● Client-side vs. Server-side technologies ● Framework & Library ● Back-End framework comparison ● Basic techniques for scaling ● What is the API?	1	*1. Laboratory work #1*
2	**Web development roadmap** ● Web development roadmap ● HTML Elements and Attributes ● HTML Forms Inputs ● Introduction to CSS ● HTML5/CSS3 features	2	*1. Laboratory work #2*
3	**JavaScript** ● JavaScript Basics ● JavaScript Standards (ES6+) ● Data Types and Variable Scoping ● Functional Programming Concepts ● Working with JSON ● DOM Manipulation ● Event Handling ● HTML Element Manipulation	3	*1. Laboratory work #3*
4	**Introduction to Angular.** ● Introduction to Angular ● Goals and Architecture of Angular ● Angular CLI usage ● JavaScript vs. TypeScript basics	4	*1. Laboratory work #4* *2. Quiz 1*
5	**Angular Components** ● Component Properties ● Data Binding ● Templates and Styles ● Life-cycle Hooks	5	*1. Laboratory work #5*
6	**Angular Modules and Router** ● Working with RESTful APIs ● Reactive Programming Concepts ● Angular Services ● Observables in Angular	6	*1. Laboratory work #6*
7	**Introduction to Python:** ● Python syntax, data types ● Sequence containers ● Functions, modules ● Object-Oriented Programming, classes, functions		*1. Laboratory work #7*
8	***Lecturers will proctor on practice lesson***		***Quiz 2 - aka Midterm***
9	**Introduction to Django:** ● What is Django?	9	*1. Laboratory work #8* *2. Project*

	- Project structure (settings.py, urls.py, wsgi.py) - Django REST Framework - Building REST APIs With Django REST Framework		
10	**Django Models & ORM** - Django models - Fields, relationships (ForeignKey, OneToOne, ManyToMany) - Migrations (makemigrations, migrate) - Admin registration - Basic ORM queries (filter, get, create)	10	1. *Laboratory work #9* 2. *Project*
11	**Building API Views (CRUD):** - DRF View types: ○ Function-Based Views (FBV) ○ Class-Based Views (CBV) - URL routing for API - Testing endpoints with Postman - HTTP methods (GET, POST, PUT, DELETE) - Response objects	11	1. *Laboratory work #10* 2. *Project*
12	**DRF Serializers and User Authentication & Permissions:** - Creating a Serializer class - Working with Serializers - Types of Serializer Classes - Simple Serializer class - ModelSerializers - Users and Authentication - Django User model - Login / Register endpoints - DRF authentication	12	1. *Laboratory work #10 - defense*
13	**Fully working Django REST API:** - Full API testing (with Postman and Angular frontend) - CORS - Selecting Objects ○ Filtering ○ Ordering ○ Slicing	13	1. *Project defense preparation*
14	**AI-Assisted Development** - Large Language Models trained to generate and understand code - AI Coding Assistants ○ GitHub Copilot ○ ChatGPT ○ Claude ○ …		***Quiz 3 - Project defense***

	- Ethical & Practical Considerations: - When and how to trust AI-generated code - Avoiding plagiarism and over-reliance - Reviewing, testing, and understanding AI outputs		
15	***Lecturers will proctor on practice lesson***		***Quiz 4 - aka Endterm***
16-17	Final Exam		

COURSE ASSESSMENT PARAMETERS

Type of Activity	Final scores
Quiz 1	5%
Quiz 2: aka Midterm	15%
Quiz 3: Project defense	10%
Quiz 4: aka Endterm	20%
Lab defense	10%
Final exam	40%
Total	**100%**

Criteria for evaluation of students during the semester:

	Assessment criteria	Weeks																Total scores
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	Quiz 1				*													5%
2	Quiz 2								*									15%
2	Quiz 3														*			10%
3	Quiz 4															*		20%
4	Lab works	*	*	*	*	*	*	*		*	*	*	*					10%
5																		
6	Final exam																*	40%
	Total																	**100%**

Academic Policy

The course follows the standard KBTU academic policy. All students are expected to uphold the highest standards of academic integrity and professional conduct.

- **Academic Integrity:**
  Cheating, duplication, data falsification, plagiarism, or the use of unauthorized materials is strictly prohibited. Any violation will result in disciplinary action according to university regulations.
- **Use of AI Tools:**
  Students are allowed to use AI-based tools (e.g., ChatGPT, GitHub Copilot, other AI assistants) outside the classroom for:
  - Learning and understanding course topics
  - Exploring examples and explanations
  - Assisting with practice and homework

  The use of AI tools is strictly prohibited during **quizzes, exams, laboratory defenses**, and any **graded in-class assessments**. Any misuse of AI during assessments will be treated as academic misconduct.
- **Attendance:**
  Attendance is **mandatory**. Students who miss more than **30%** of classes will receive an automatic **F (Fail)** grade for the course.
  Merely attending class does not constitute participation. Students are expected to come prepared, engage actively in discussions.
- **Class Participation:**
  Participation includes:
  - Reading assigned materials in advance
  - Engaging in class discussions
  - Asking relevant questions
  - Demonstrating consistent effort in labs and projects
- **Communication:**
  MS Teams is the primary platform for announcements, communication, and file sharing. Students must **check MS Teams channels of the course and messages daily** and respond promptly when needed.
- **Assignments and Deadlines:**
  All written work must be typed or clearly handwritten and submitted by the specified deadline.
  Late submissions are not accepted.
- **Punctuality:**
  Students must arrive on time. Repeated lateness will be treated as absenteeism.
- **Missed Work and Make-Up Tests:**
  Students are responsible for catching up on any missed work.
  Make-up tests are not guaranteed and will only be considered in exceptional cases, subject to instructor approval.
- **Classroom Etiquette:**
  - Mobile phones must be **switched off or silenced** during class.
  - Respectful and professional behavior toward instructors and peers is required at all times.