KAZAKH-BRITISH TECHNICAL UNIVERSITY

SYLLABUS

Academic Year 2024-2025
Semester 1

Discipline: *Backend for High Loaded Environment*

Volume of credits: 5

Course/Syllabus designer: *Azamat Serek, Assistant-Professor*

Instructor's e-mail	a.serek@kbtu.kz
Class Schedule	Monday (11:00 - 13:00 lectures). Friday (15:00-17:00 practices)
Office Hours	Saturday 09:00-18:00
Office	

1. Course Overview

This course is designed to equip students with the skills and knowledge necessary to build and manage robust backend systems capable of handling large volumes of traffic and data. Throughout the course, students will explore the core concepts of high-load systems, including database design, caching strategies, load balancing, and distributed systems. Emphasis is placed on practical techniques for optimizing performance, ensuring data consistency, and maintaining system resilience in high-load environments. The course also covers modern practices in monitoring, security, and continuous integration/deployment, providing a comprehensive understanding of backend systems' challenges and solutions.

Prerequisites: _____

Postrequisites: _____

2. Aims and objectives

The primary aim of this course is to equip students with the expertise needed to design, build, and manage backend systems capable of handling high traffic and large data volumes. The

course seeks to develop a strong foundation in the fundamental principles and advanced techniques necessary for creating scalable, reliable, and efficient backend systems.

Objectives:

1. To provide a comprehensive understanding of high-load systems and their unique challenges.
2. To develop students' ability to design and optimize databases specifically for high-load environments.
3. To teach effective caching strategies that improve system performance and reduce load times.
4. To explore various load balancing techniques to distribute traffic evenly and maintain system stability.
5. To enhance knowledge of distributed systems, focusing on maintaining data consistency and ensuring efficient communication between components.
6. To guide students in scaling backend systems to accommodate growing demands while preserving performance and reliability.
7. To introduce modern monitoring and observability practices that allow for real-time system performance tracking and issue resolution.

3. Learning outcomes

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

1. Design and optimize databases for high-load systems.
2. Implement effective caching and load balancing strategies.
3. Understand and apply principles of distributed systems and data consistency.
4. Scale backend systems to meet growing demands.
5. Monitor and optimize system performance in real-time.
6. Integrate security practices into backend system design.
7. Ensure fault tolerance and resilience in high-load environments.
8. Apply testing and CI/CD practices to maintain system reliability and efficiency.

4.Textbooks and readings

Primary textbook

1. "Two Scoops of Django 3.x: Best Practices for the Django Web Framework" by Daniel Roy Greenfeld and Audrey Roy Greenfeld
2. Building secure and reliable systems:best practices for designing, implementing, and maintaining systems / Adkins Heather [et al.]. - USA : O'REILLY, 2020. - p.519 : ill. - ISBN 978-1492-08312-2 : 28000-00. Authors: Adkins Heather, Beyer Betsy, Blankinship Piotr, Oprea Ana, Stubblefield Adam

Supplementary textbooks

1. "High Performance Django" by Peter Baumgartner and Yevgeniy Brikman This book focuses on performance optimization and scaling Django applications, offering practical insights and strategies for managing high-load systems.
2. "Django for APIs: Build web APIs with Python and Django" by William S. Vincent This resource delves into Django REST Framework, providing in-depth coverage of API design, serializers, views, and permissions essential for developing robust APIs.

3. "Mastering Django: Core" by Nigel George
   An in-depth exploration of Django's core features, including advanced authentication, security practices, and configuration techniques relevant to high-load environments.
4. "Django 4 By Example" by Antonio Mele
   This book presents practical Django projects and scenarios, including performance tuning and optimization, providing hands-on examples and solutions for high-load applications.

5. Lesson Program

Week	Classes				
	Topic	**Lecture**	**Laboratory**	**Tutorial**	**Textbook Chapter**
1	Introduction to High-Load Systems	2	1		1
2	Backend Fundamentals	2	1		2
3	Database Design and Optimization	2	1		3
4	Caching Strategies	2	1		4
5	Load Balancing Techniques	2	1		5
6	Distributed Systems and Data Consistency	2	1		6
7	Midterm	2	1		7
8	Scaling Backend Systems	2	1		-
9	Monitoring and Observability	2	1		8
10	Performance Tuning and Optimization	2	1		9
11	Message Queues and Asynchronous Processing	2	1		10
12	Security in High-Load Systems	2	1		11
13	Fault Tolerance and Resilience	2	1		12
14	Testing and Continuous Integration/Continuous Deployment (CI/CD)	2	1		13
15	Review	2	1		14

6. Course Requirements and Grades

COURSE ASSESSMENT PARAMETERS

Assignments	30%

Midterm	30%
Final exam	40 %
Total	**100 %**

№	Assessment criteria	Weeks																Total
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 17	
1	Activity on seminars and homework			+			+				+			+				20 %
2	Quizes and mid/end-terms							+										40 %
3	Final examination																+	40 %
	Total	60															40	100 %

Grading policy:

1. 4 Assignments (30)
2. Midterm (30)
3. Final (40)

Grade		Achievement percentage	Assessment criterion
«Excellent»	A	95-100%	This grade is given when the student: demonstrated a complete understanding of the course material; did not make any errors or inaccuracies; completed control and laboratory work in a timely and correct manner, and submitted reports on them; demonstrated original thinking; submitted control quizzes on time and without any errors; completed homework assignments; engaged in research work; independently used additional scientific literature in studying the discipline; was able to independently systematize the course material.
	A -	90-94%	
«Good»	B+	85-89%	This grade is given when the student: Has mastered the course material at no less than 75%; Did not make gross errors in responses; Timely completed control and laboratory work and submitted them without fundamental remarks;
	B	80-84%	

	B-	75-79%	Correctly completed and timely submitted control tests and homework assignments without fundamental remarks; Utilized additional literature as indicated by the instructor; Engaged in research work, made non-fundamental errors, and fundamental errors corrected by the student themselves; Managed to systematize the course material with the help of the instructor.
	C+	70-74%	
«Satisfactory»	C	65-69%	This grade is given when the student: Has mastered the course material no less than 50%; Required assistance from the instructor when completing control and laboratory work, homework assignments; Made inaccuracies and non-fundamental errors when submitting control tests; Did not demonstrate activity in research work, relied solely on the educational literature indicated by the instructor; Experienced more difficulty in systematizing the material.
	C-	60-64%	
	D+	55-59%	
	D	50-54%	

ATTENTION!

1) If student missed more than **30% of lessons student receives «F (Fail)»** grade;

2) If for two attestations student receives 29 or less points, this student is not accepted to final exam and for all course he (she) receives **«F (Fail)» grade;**

3) If student receives on final exam 19 or less points, then independently on how many points he (she) received for two attestations, in whole he (she) receives **«F (Fail)» grade;**

In the case of missing or being late for final exam without plausible reason, independently on how many points he (she) received for two attestations, in whole he (she) receives **«F (Fail)» grade.**

4) If a student missed more than 30% of the lectures due to health problems and has medical documents in the form of KBTU, but did not complete the course and it is recommended to take an academic leave.

5) In case of detection **of plagiarism** in the course, the student is automatically receives **«F (Fail)» grade**.

Academic Policy:

- Cheating, duplication, falsification of data, plagiarism are not permitted under any circumstances!
- Students must participate fully in every class. While attendance is crucial, merely being in class does not constitute "participation". Participation means reading the assigned materials, coming to class prepared to ask questions and engage in discussion.
- Students are expected to take an active role in learning (the instructor will provide the information and guidelines to do this).
- Students must come to class on time.
- Students are to take responsibility for making up any work missed.
- Make up tests in case of absence will not normally be allowed.
- Mobile phones must always be switched off in class.
- Students should always show tolerance, consideration and mutual support towards other students.

Students are encouraged to

- consult the teacher on any issues related to the course;
- make up within a week's time for the works undone for a valid reason without any grade deductions;
- make any proposals on improvement of the academic process;
- track down their continuous rating throughout the semester.

Minutes #1 of School of Information Technology and Engineering meeting on August 20, 2024