



SYLLABUS

Academic Year 2024-2025
Semester 1

Discipline: *Golang application development*

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

The course commences with a detailed exploration of Golang's syntax and foundational principles. As participants progress through the course, they will delve into the intricacies of Object-Oriented Programming (OOP) in Golang, gaining insights into its unique code structuring approach.

A significant emphasis is placed on database interaction, covering two segments. The first ensures proficiency in fundamental database operations, while the subsequent part delves into advanced techniques, equipping participants to navigate complex data scenarios. The course seamlessly transitions into the domain of RESTful API design, imparting essential skills for crafting robust APIs to facilitate seamless communication between the backend and frontend.

Building on this knowledge, students explore the nuances of connecting their Golang backend with the frontend in two comprehensive parts. Authentication and Authorization assume a pivotal role, guiding participants in implementing secure mechanisms to safeguard user data and manage access effectively.

The course unfolds the concurrent and parallel capabilities of Golang, offering insights into efficient task handling. Security considerations take precedence as participants learn to fortify their applications against common web vulnerabilities and ensure secure communication. Logging and monitoring practices are explored to maintain application health and performance. Caching mechanisms are introduced to optimize data retrieval, enhancing application efficiency. Dependency management skills are honed, enabling participants to seamlessly integrate external packages using Go modules or other tools. The course concludes with a two-part exploration of deployment strategies, covering Docker, Kubernetes, and cloud services to equip participants with the skills needed for scalable and efficient application deployment.

Prerequisites: _____

Postrequisites: _____

2. Aims and objectives

Aims:

- To introduce students to the Go programming language (Golang) and its core features.
- To develop students' ability to design, build, and maintain scalable and efficient applications using Golang.
- To provide hands-on experience in developing full-stack applications with Golang as the backend.
- To explore advanced topics such as concurrency, security, and containerization in the context of Golang development.

Objectives:

- To understand the basics of Go syntax, data structures, and programming paradigms.
- To explore object-oriented programming (OOP) principles in Golang.
- To learn how to connect Go applications to databases and perform CRUD operations.
- To design and implement RESTful APIs using Golang.
- To integrate Golang backend services with frontend frameworks.
- To develop a solid understanding of authentication, authorization, and security best practices in Golang.
- To explore concurrency, parallelism, and their implementation in Go.
- To learn techniques for logging, monitoring, caching, and containerizing Go applications.

3. Learning outcomes

By the end of the course, students will be able to:

1. Write efficient and idiomatic Go code, leveraging the language's core syntax and features.
2. Apply object-oriented programming principles within the Go ecosystem.
3. Connect Go applications to databases, handling data storage and retrieval operations.
4. Design and develop RESTful APIs in Golang.
5. Implement authentication and authorization mechanisms in Go applications.
6. Understand and apply concurrency and parallelism techniques in Go.

7. Ensure security, logging, monitoring, and caching in Go-based projects.
8. Containerize Go applications using Docker and other relevant tools.

4. Textbooks and readings

Primary textbook

1. **"The Go Programming Language"** by Alan A. A. Donovan and Brian W. Kernighan. Addison-Wesley, 2015.

Supplementary textbooks

1. **"Go in Action"** by William Kennedy, Brian Ketelsen, and Erik St. Martin. Manning Publications, 2015.
Schmitt Christopher.
2. Designing Web And Mobile Graphics Fundamental Concepts for WEB and interactive projects. - USA : New Riders, 2013. - 301 с. : илл. - ISBN 978-0-321-85854-2 : 17900-00.

5. Lesson Program

Week	Classes				
	Topic	Lecture	Laboratory	Tutorial	Textbook Chapter
1	Intro to Go. Basics of syntax.	2	1		1
2	OOP in Golang.	2	1		2
3	Dependency management.	2	1		3
4	Working with database, part 1	2	1		4
5	Working with database, part 2.	2	1		5
6	Connecting with frontend, part 1.	2	1		6
7	Midterm	2	1		7
8	Connecting with frontend, part 2.	2	1		-
9	Authentication and authorization.	2	1		8
10	Concurrency and Parallelism.	2	1		9
11	Security.	2	1		10
12	Logging and monitoring	2	1		11
13	Caching	2	1		12
14	Containerizing	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 7		
1.	Attendance has to be more than 80%																		
2.	Activity on seminars and homework			+			+				+			+				20%	
4.	Quizzes and mid/end-terms								+									40%	
5.	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%	<p>This grade is given when the student:</p> <p>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.</p>
	A -	90-94%	

«Good»	B+	85-89%	<p>This grade is given when the student:</p> <p>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; 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.</p>
	B	80-84%	
	B-	75-79%	
	C+	70-74%	
«Satisfactory»	C	65-69%	<p>This grade is given when the student:</p> <p>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.</p>
	C-	60-64%	
	D+	55-59%	
	D	50-54%	

7. General Rules

Academic Integrity

Attendance policy

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