


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Відокремлений структурний підрозділ
«Педагогічний фаховий коледж
Львівського національного університету імені Івана Франка»

ЗАТВЕРДЖЕНО

На засіданні циклової комісії
спеціальності "Комп'ютерні науки"
протокол № 1 від «30» серпня 2024 р.

Голова циклової комісії

 Олександр КНИГІНЦЬКИЙ

СИЛАБУС З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«Моніторинг роботи програмних систем і комплексів»

Освітньо-професійний ступінь	«фаховий молодший бакалавр»
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітньо-професійна програма	Комп'ютерні науки
Форма навчання	денна

Назва дисципліни	Моніторинг роботи програмних систем і комплексів
Адреса викладання дисципліни	м. Львів, вул. Тарнавського, 107
Факультет та кафедра, за якою закріплена дисципліна	Відокремлений структурний підрозділ «Педагогічний фаховий коледж Львівського національного університету імені Івана Франка». Циклова комісія спеціальності Комп'ютерні науки
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 122 Комп'ютерні науки
Викладачі дисципліни	Майорак Богдан Ярославович
Контактна інформація викладачів	maiorchakbohdan@gmail.com
Консультації з питань навчання по дисципліні відбуваються	Консультації проводяться в день проведення лекційних занять (за попередньою домовленістю) у корпусі по вул. Ген. Тарнавського, 107, Ауд.303. В режимі он-лайн: консультації проводяться за домовленістю з студентами чи групою студентів на платформі MS Teams та на інших електронних ресурсах, за попереднім погодженням часу (через електронну пошту викладача або телефонний дзвінок).
Сторінка дисципліни	https://pedcollege.lnu.edu.ua/course/monitorung-programnuh-sistem/
Інформація про дисципліну	Дисципліна «Моніторинг роботи програмних систем і комплексів» є вибірковою освітньою компонентом ОПП циклу професійної та практичної підготовки зі спеціальності 122 Комп'ютерні науки для освітньої програми «Комп'ютерні науки», яка викладається в 6 семестрі в обсязі 3-х кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальна дисципліна «Моніторинг роботи програмних систем і комплексів» спрямована на формування у студентів знань та вмінь щодо методів контролю, аналізу та оптимізації роботи програмного забезпечення та ІТ-інфраструктури. Вона охоплює принципи збору, обробки та візуалізації метрик продуктивності, використання сучасних засобів логування та автоматизованого оповіщення про збої. Особлива увага приділяється інструментам моніторингу, методам аналізу навантаження, забезпеченню безперервності роботи систем та підвищенню їхньої надійності в умовах високої динаміки сучасних ІТ-середовищ.
Мета та цілі дисципліни	Мета: Надати студентам глибокі знання про методи моніторингу програмних систем і комплексів, принципи збору, аналізу та обробки даних про продуктивність і стабільність роботи програмного забезпечення. Студенти навчатимуться використовувати сучасні засоби моніторингу, логування та алертингу для забезпечення надійності та ефективного управління ІТ-інфраструктурою. Цілі: <ul style="list-style-type: none"> • Ознайомити студентів із принципами моніторингу програмних систем, методами збору метрик та їх аналізу.

	<ul style="list-style-type: none"> • Вивчити сучасні інструменти моніторингу (Prometheus, Grafana, AWS CloudWatch, Azure Monitor, ELK Stack) та їхнє застосування. • Навчити працювати з системами логування та алертингу, автоматизувати процеси виявлення та усунення проблем. • Ознайомити з методами аналізу навантаження, виявлення аномалій та оптимізації продуктивності програмних комплексів. • Розглянути підходи до моніторингу в DevOps-процесах, інтеграцію з CI/CD та автоматизоване управління інцидентами.
<p>Література для вивчення дисципліни</p>	<p>Основна література:</p> <ol style="list-style-type: none"> 1. Добришин Ю. Є. "Адміністрування програмних систем і комплексів. Методичні вказівки щодо виконання лабораторних робіт." Київ: Університет економіки та права «КРОК», 2023. library.krok.edu.ua 2. Лавріщева К. М. "Програмна інженерія." Київ: Академперіодика, 2008. college.suitt.edu.ua+1csc.knu.ua+1 3. Коваленко О. С., Добровська Л. М. "Проектування інформаційних систем: Загальні питання теорії проектування ІС." Конспект лекцій. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 2020. ela.kpi.ua+1ela.kpi.ua+1 4. Лавріщева К. М. "Інститут програмних систем." Монографія. Київ, 2011. library.krok.edu.ua+7csc.knu.ua+7college.suitt.edu.ua+7 5. Авторський колектив. "Тестування програмних систем і комплексів." Навчальний посібник. Київ: СУІТТ, 2023.
<p>Обсяг курсу</p>	<p>60 години аудиторних занять. З них 36 години лекцій, 24 години практичних робіт на кожну із 8 бригад та 30 годин самостійної роботи</p>
<p>Очікувані результати навчання</p>	<p>В результаті вивчення даного курсу студент повинен:</p> <p>Знати:</p> <ul style="list-style-type: none"> • Принципи моніторингу програмних систем, методи збору, обробки та аналізу метрик продуктивності. • Архітектуру та функціональні можливості систем моніторингу, зокрема Prometheus, Grafana, AWS CloudWatch, Azure Monitor, ELK Stack. • Основи логування та аналізу журналів подій, методи трасування запитів у розподілених системах. • Принципи автоматичного алертингу, управління інцидентами та аналізу причин несправностей. <p>Вміти:</p> <ul style="list-style-type: none"> • Налаштовувати та використовувати сучасні системи моніторингу для збору та аналізу метрик продуктивності ІТ-систем. • Розгортати та конфігурувати засоби логування та трасування у програмних комплексах. • Автоматизувати процеси алертингу та виявлення несправностей, використовуючи DevOps-інструменти. • Аналізувати навантаження, виявляти аномалії та оптимізувати продуктивність програмних систем. • Інтегрувати моніторинг у CI/CD-процеси для забезпечення безперервного контролю якості програмного забезпечення.

Загальні компетентності	ЗК4. Здатність застосовувати знання у практичних ситуаціях. ЗК7. Здатність спілкуватися іноземною мовою. ЗК8. Здатність вчитися і оволодівати сучасними знаннями.
Спеціальні (фахові) компетентності	ФК3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища. ФК4. Здатність здійснювати проектування та розробку програмного забезпечення. ФК9. Здатність застосовувати знання сучасних методів і технологій створення та супроводження розподілених систем.
Програмні результати навчання	ПРН05. Розуміти основні методи і технології об'єктно-орієнтованого та компонентного програмування. ПРН10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення. ПРН11. Застосовувати сучасні мови програмування та технології для розробки програмного забезпечення розподілених систем. ПРН12. Знати основні принципи функціонування системного та прикладного програмного забезпечення.

Ключові слова	Моніторинг програмних систем. Метрики продуктивності. Логування. Алертинг. Трасування. Аналіз журналів подій. Візуалізація метрик. Відмовостійкість. Автоматизоване управління інцидентами. CI/CD інтеграція. DevOps-моніторинг. Prometheus. Grafana. ELK Stack. AWS CloudWatch. Azure Monitor. Google Cloud Operations. Системи логування. Аналіз навантаження. Оптимізація продуктивності. Резервне копіювання. Відновлення даних. Тестування надійності.
Формат курсу	Очний
	Проведення лекцій, практичних робіт та консультації для кращого розуміння тем
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік в кінці семестру в усній та практичній формі
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування», «Операційні системи» та «Цифрова схемотехніка».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Лекції на основі презентацій, лабораторні роботи на основі методичок та їхній захист на основі індивідуальних звітів, виконання індивідуальних завдань та їхній захист.
Необхідне обладнання	Пристрої мультимедіа. ОС Windows 11, Visual Studio 2022, платформи MS Teams.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання знань студентів здійснюється за 100-бальною шкалою. Максимальна кількість балів при оцінюванні знань впродовж семестру становить 100 балів. Програмою передбачено дві модульні контрольні роботи. Студент за одну контрольну роботу може отримати 25 балів (25x2 = 50).

Програмою передбачено 18 практично-семінарських занять. За практичне заняття - від 1 до 2 балів . Кількість балів, які студент може отримати за усні відповіді подано в таблиці.

За **самостійну роботу** впродовж семестру студент набирає 20 балів.

Критерії оцінювання результатів навчання

90 – 100 б. – здобувач повно та ґрунтовно засвоїв всі теми навчальної програми, вміє вільно викладати зміст, має глибокі, міцні, систематичні знання всіх питань навчальної дисципліни, розуміє їх значення для своєї професійної підготовки; виконав усі завдання кожної теми та модульного поточного контролю.

81 – 89 б. - здобувач ґрамотно і по суті викладає програмний матеріал, застосовує теоретичні знання при виконанні практичних завдань, однак допускає незначні неточності, засвоїв більшість тем навчальної програми, вміє самостійно викладати зміст всіх питань навчальної дисципліни; виконав завдання кожної теми та модульного поточного контролю.

71 – 80 б. - здобувач добре знає програмний матеріал, володіє базовими навичками з виконання практичних завдань, самостійно обирає метод реалізації, але не завжди здатний провести аналіз і узагальнення результату, не завжди вміє вільно викладати зміст всіх питань навчальної дисципліни; виконав більшість завдань кожної теми та модульного поточного контролю.

61 – 70 б. – здобувач засвоїв тільки основний матеріал на рівні репродуктивного відтворення, але не знає окремих деталей, припускається неточностей, порушує послідовність у викладі матеріалу, може розв'язувати типові завдання за зразком, але допускає помилки, виконав окремі завдання кожної теми та модульного поточного контролю.

51 - 60 б. - здобувач відтворює основні поняття і визначення курсу, але досить поверхово, не виділяючи взаємозв'язок між ними, може сформулювати за допомогою викладача основні тези теми, допускає помилки, які повною мірою самостійно виправити не може; відчуває труднощі під час виконання практичних завдань, виконав лише деякі завдання кожної теми та модульного контролю.

	<p>21 – 50 б. – здобувач не знає значної частини програмного матеріалу, допускає суттєві помилки, з великими труднощами виконує практичні завдання, знайомий лише з деякими поняттями та визначеннями курсу; не виконав практичні завдання та завдання модульного поточного контролю.</p> <p>0 – 20 б. - необхідний повторний курс з навчальної дисципліни.</p>
Питання до контрольних робіт	Білету для підсумкового заліку студенти отримують на останній лекції курсу.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Номер	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	<p>Вступ до моніторингу програмних систем. Основні поняття та термінологія. Необхідність моніторингу</p> <p>Тема знайомить студентів із основами моніторингу програмних систем, його роллю у забезпеченні стабільної та ефективної роботи програмного забезпечення. Розглядаються ключові поняття та термінологія моніторингу, включаючи метрики, логування, трасування та алертинг. Аналізується необхідність моніторингу для виявлення проблем, оптимізації продуктивності та забезпечення відмовостійкості систем. Особливу увагу приділяється видам моніторингу (інфраструктурний, прикладний, мережевий) та їхньому значенню у сучасних ІТ-екосистемах.</p>	Лекція	1, 2, 3, 4, 5	Збір та аналіз логів за допомогою ELK Stack (Elasticsearch, Logstash, Kibana)	<p>I -</p> <p>03.03 10.03</p> <hr/> <p>II -</p> <p>03.03 10.03</p>
2	<p>Архітектура програмних систем та їхні вузькі місця. Компоненти сучасних ПЗ. Основні проблеми в продуктивності .</p> <p>Тема знайомить студентів із основними архітектурними компонентами сучасного програмного забезпечення та їхнім впливом на продуктивність систем. Розглядаються ключові елементи, такі як сервери додатків, бази даних, мережеві взаємодії та системи кешування. Аналізуються основні</p>	Лекція	1, 2, 3, 4, 5	Збір та аналіз логів за допомогою ELK Stack (Elasticsearch, Logstash, Kibana)	<p>I -</p> <p>11.03 14.03</p> <hr/> <p>II -</p> <p>11.03 14.03</p>

	вузькі місця в продуктивності, зокрема перевантаження ресурсів, блокування процесів, неефективне використання пам'яті та проблеми з масштабованістю. Окрема увага приділяється методам виявлення та усунення цих проблем за допомогою сучасних засобів моніторингу.				
3	<p>Метрики та показники ефективності програмних систем. Час відгуку, використання ресурсів. Помилки та винятки</p> <p>Тема знайомить студентів із ключовими метриками ефективності програмних систем та їхньою роллю в оцінці продуктивності. Розглядаються основні показники, зокрема час відгуку, пропускна здатність, використання процесора, пам'яті, дискових і мережевих ресурсів. Аналізуються методи збору та інтерпретації метрик для виявлення потенційних проблем у роботі програмного забезпечення. Окрема увага приділяється моніторингу помилок і винятків, їхній класифікації, аналізу причин виникнення та засобам автоматичного виявлення і сповіщення про збої.</p>	Лекція	1, 2, 3, 4, 5	Збір та аналіз логів за допомогою ELK Stack (Elasticsearch, Logstash, Kibana)	<p>I - 15.03 18.03</p> <hr/> <p>II - 15.03 18.03</p>
4	<p>Засоби збору логів та аналіз логів. Типи логів. Інструменти для збору та обробки.</p> <p>Тема знайомить студентів із процесом збору, обробки та аналізу логів у програмних системах. Розглядаються типи логів, зокрема системні, додаткові, мережеві, безпекові та аплікаційні логи, їхня структура та значення для діагностики роботи програмного забезпечення. Аналізуються методи централізованого збору логів, форматування та збереження, а також найкращі практики обробки логів. Окрема увага приділяється популярним інструментам збору та аналізу логів, таким як ELK Stack (Elasticsearch, Logstash, Kibana), Graylog, Splunk, Fluentd та їхньому застосуванню для моніторингу та виявлення аномалій у програмних системах.</p>	Лекція	1, 2, 3, 4, 5	Моніторинг серверних додатків за допомогою Prometheus і Grafana	<p>I - 21.03 24.03</p> <hr/> <p>II - 21.03 24.03</p>
5	<p>Моніторинг серверних додатків. Методи контролю навантаження. Моніторинг процесів та служб</p>		1, 2, 3, 4	Моніторинг серверних додатків	

	<p>Тема знайомить студентів із методами моніторингу серверних додатків для забезпечення їхньої стабільної та продуктивної роботи. Розглядаються основні метрики продуктивності серверів, зокрема використання процесора, пам'яті, дискового простору, мережевого трафіку. Аналізуються методи контролю навантаження, включаючи балансування ресурсів, автоматичне масштабування та оптимізацію запитів. Окрема увага приділяється моніторингу процесів і служб, їхньому статусу, часу виконання, споживанню ресурсів та автоматичному відновленню в разі збоїв. Розглядаються інструменти моніторингу, такі як Prometheus, Grafana, Zabbix, Nagios, New Relic, та їх застосування у сучасних IT-інфраструктурах.</p>			за допомогою Prometheus і Grafana	<p>I - 25.03 28.03</p> <hr/> <p>II - 25.03 28.03</p>
6	<p>Моніторинг баз даних. Показники продуктивності БД. Аналіз запитів та оптимізація. Тема знайомить студентів із методами моніторингу баз даних для забезпечення їхньої стабільної та ефективної роботи. Розглядаються ключові показники продуктивності, такі як час відгуку запитів, використання процесора та пам'яті, рівень блокувань та кешування, кількість з'єднань. Аналізується вплив неправильно складених запитів на продуктивність БД та методи їх оптимізації.</p>	Лекція	1, 2, 3, 4	Моніторинг серверних додатків за допомогою Prometheus і Grafana	<p>I - 01.04 04.04</p> <hr/> <p>II - 01.04 04.04</p>
7	<p>Моніторинг веб-додатків. Навантаження веб-серверів. Моніторинг API. Тема знайомить студентів із методами моніторингу веб-додатків для забезпечення їхньої стабільності, продуктивності та безпеки. Розглядаються основні метрики моніторингу, такі як час відгуку сервера, кількість активних з'єднань, використання ресурсів, рівень помилок (HTTP 4xx, 5xx). Аналізується навантаження на веб-сервери та методи його балансування за допомогою NGINX, Apache, Load Balancers. Окрема увага приділяється моніторингу API, аналізу продуктивності запитів, виявленню помилок та перевірці доступності сервісів за допомогою Postman, Prometheus, Grafana, AWS</p>	Лекція	1, 2, 3, 4	Моніторинг баз даних MySQL/PostgreSQL: аналіз продуктивності запитів	<p>I - 07.04 08.04</p> <hr/> <p>II - 07.04 08.04</p>

	API Gateway Monitoring. Вивчаються підходи до стрес-тестування, виявлення вузьких місць та автоматизованого алертингу.				
8	<p>Моніторинг мережі та мережевої безпеки. Аналіз трафіку мережі. Виявлення аномалій.</p> <p>Тема знайомить студентів із методами моніторингу мережевої інфраструктури для забезпечення стабільності, продуктивності та безпеки з'єднань. Розглядаються ключові показники мережевого моніторингу, такі як пропускна здатність, затримка, втрата пакетів, використання каналів зв'язку та рівень навантаження на мережеві пристрої</p>	Лекція	1, 2, 3, 4	<p>Моніторинг контейнерів Docker та Kubernetes з використанням Agro/Lens</p> <p>Модульна контрольна робота 1.</p>	<p>I - 11.04 15.04</p> <hr/> <p>II - 11.04 15.04</p>
9	<p>Моніторинг контейнеризованих додатків (Docker, Kubernetes). Особливості моніторингу контейнерів. Інструменти спостереження.</p> <p>Тема знайомить студентів із підходами до моніторингу контейнеризованих додатків, що працюють у середовищах Docker та Kubernetes. Розглядаються особливості моніторингу контейнерів, такі як динамічне масштабування, короткотривалість життєвого циклу, управління ресурсами (CPU, RAM, диск, мережа), моніторинг стану контейнерів та подій. Аналізуються методи збору метрик, виявлення помилок та аналіз логів у контейнеризованих середовищах. Окрема увага приділяється популярним інструментам спостереження, зокрема Prometheus, Grafana, cAdvisor, ELK Stack, Fluentd, Kubernetes Metrics Server, Loki, а також механізмам автоматизованого алертингу та масштабування у Kubernetes.</p>	Лекція	1, 2, 3, 4	<p>Моніторинг контейнерів Docker та Kubernetes з використанням Agro/Lens</p>	<p>I - 18.04 21.04</p> <hr/> <p>II - 18.04 21.04</p>
10	<p>Автоматизація моніторингу. CI/CD поняття та моніторинг. Автоматизовані дашборди.</p> <p>Тема знайомить студентів із методами автоматизації моніторингу програмних систем для забезпечення безперервного контролю продуктивності та швидкого реагування на інциденти. Розглядається взаємозв'язок моніторингу з CI/CD-процесами, включаючи вбудований моніторинг у конвеєрі розгортання, тестування</p>	Лекція	1, 2, 3, 4	<p>Моніторинг хмарних сервісів (AWS, Azure, Google Cloud)</p>	<p>I - 22.04 25.04</p> <hr/> <p>II - 22.04 25.04</p>

	продуктивності та аналіз помилок. Аналізуються інструменти автоматизації моніторингу, такі як Prometheus, Grafana, ELK Stack, AWS CloudWatch, Azure Monitor, та їхня інтеграція з DevOps-платформами (Jenkins, GitLab CI/CD, GitHub Actions)				
11	Моніторинг хмарних сервісів. AWS CloudWatch, Azure Monitor. Тема знайомить студентів із принципами моніторингу хмарних сервісів для забезпечення продуктивності, доступності та безпеки IT-інфраструктури. Розглядаються основні метрики та показники ефективності у хмарному середовищі, зокрема навантаження на віртуальні машини, використання ресурсів, мережевий трафік, логи додатків та безпекові події. Аналізуються можливості AWS CloudWatch та Azure Monitor для збору, аналізу та візуалізації метрик, а також налаштування автоматизованих алертів та реагування на інциденти.	Лекція	1, 2, 3, 4	Моніторинг веб-серверів (Apache, Nginx) та їхнє навантаження	I - 29.04 02.05 29.04 II - 02.05
12	Моніторинг, логування та алертинг у хмарі. Сервіси моніторингу (AWS CloudWatch, Azure Monitor, GCP Monitoring). Лекція присвячена механізмам моніторингу, логування та алертингу у хмарних середовищах, що забезпечують стабільну роботу та оперативне реагування на інциденти. Розглядаються принципи збору, аналізу та візуалізації метрик для оцінки продуктивності хмарних сервісів. Аналізуються сервіси моніторингу, такі як AWS CloudWatch, Azure Monitor, GCP Monitoring, а також їхні можливості щодо аналізу логів, виявлення аномалій та автоматизованих сповіщень (алертингу)	Лекція	1, 2, 3, 4	Моніторинг веб-серверів (Apache, Nginx) та їхнє навантаження Модульна контрольна робота 2.	I - 05.05 06.05 05.05 II - 06.05
13	Виявлення аномалій у роботі програмних систем. Методи машинного навчання. Використання статистичних моделей. Тема знайомить студентів із сучасними підходами до виявлення аномалій у роботі програмних систем, що дозволяють ідентифікувати відхилення від нормальної поведінки та запобігати можливим збоєм. Розглядаються основні методи машинного	Лекція	1, 2, 3, 4	Автоматизація моніторингу з використанням Python (наприклад, написання власного бота для Telegram-оповіщення)	I - 09.05 13.05 09.05 II - 13.05

	навчання, такі як кластеризація (k-means), нейронні мережі, ізольований ліс (Isolation Forest), алгоритми прогнозування на основі часових рядів (ARIMA, LSTM).				
14	<p>Моніторинг мобільних додатків. Особливості та основні підходи.</p> <p>Тема знайомить студентів із методами моніторингу мобільних додатків для забезпечення їхньої стабільності, продуктивності та зручності використання.</p> <p>Розглядаються ключові метрики, зокрема час відгуку, споживання ресурсів (CPU, пам'ять, батарея), рівень аварійних завершень (crash rate), продуктивність мережевих запитів. Аналізуються особливості моніторингу мобільних додатків у порівнянні з веб- та серверними рішеннями, включаючи збір логів із пристроїв, тестування на різних платформах (iOS, Android) та підтримку різних мережевих умов</p>	Лекція	1, 2, 3, 4, 5	Автоматизація моніторингу з використанням Python (наприклад, написання власного бота для Telegram-оповіщення)	<p>I - 16.05 19.05</p> <hr/> <p>II - 16.05 19.05</p>
15	<p>Моніторинг у реальному часі та аналіз даних. Поточкова обробка. Реагування на інциденти.</p> <p>Тема знайомить студентів із принципами моніторингу в реальному часі, що дозволяє оперативно відстежувати стан програмних систем та миттєво реагувати на критичні події.</p> <p>Розглядаються основи потокової обробки даних, методи збору та аналізу великих потоків інформації за допомогою Apache Kafka, Apache Flink, Spark Streaming. Аналізуються підходи до автоматичного виявлення відхилень та генерації алертів за допомогою Prometheus, Grafana, AWS CloudWatch, Azure Monitor.</p>	Лекція	1, 2, 3, 4, 5	Автоматизація моніторингу з використанням Python	<p>I - 20.05 23.05</p> <hr/> <p>II - 20.05 23.05</p>
16	<p>Кібербезпека та моніторинг атак. Виявлення DDoS атак. Моніторинг логів безпеки.</p> <p>Тема знайомить студентів із методами забезпечення кібербезпеки шляхом моніторингу та аналізу загроз у програмних системах.</p> <p>Розглядаються основні види атак, такі як DDoS, brute-force, SQL-ін'єкції, XSS та методи їхнього виявлення. Аналізуються підходи до моніторингу логів безпеки, виявлення підозрілої активності та попередження атак за допомогою SIEM-систем (Splunk, Wazuh, IBM QRadar, ArcSight).</p>	Лекція	1, 2, 3, 4, 5	Реалізація та захист підсумкового проекту.	<p>I - 23.05 27.05</p> <hr/> <p>II - 23.05 27.05</p>

17	<p>Розробка стратегій моніторингу. Планування та імплементація стратегій. Аналіз реальних програм.</p> <p>Тема знайомить студентів із принципами розробки ефективних стратегій моніторингу для різних типів програмних систем. Розглядаються основні етапи планування моніторингу, зокрема визначення ключових метрик, вибір інструментів збору даних, налаштування алертингу та автоматизації процесів. Аналізуються стратегії моніторингу для веб-додатків, мікросервісних архітектур, баз даних, мережевої інфраструктури та контейнеризованих рішень.</p>	Лекція	1, 2, 3, 4	Реалізація та захист підсумкового проекту.	<p>I - 27.05</p> <p>27.05</p> <hr/> <p>27.05</p> <p>II - 27.05</p>
18	<p>Підведення підсумків курсу</p> <p>Обговорення екзаменаційних білетів.</p>	Залік		Опрацювання білетів	I; II - 30.05