

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПРАКТИКУМ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю
151 «Автоматизація та комп'ютерно-інтегровані технології»,
освітньо-професійною програмою
«Технічні та програмні засоби автоматизації»*

Київ
КПІ ім. Ігоря Сікорського
2022

Технології розроблення програмного забезпечення: практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», освітньо-професійна програма «Технічні та програмні засоби автоматизації» / КПІ ім. Ігоря Сікорського; уклад.: Д. О. Ковалюк. – Електронні текстові данні (1 файл: 3,2 Мбайт). - Київ : КПІ ім. Ігоря Сікорського, 2022. – 55 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол №3 від 01.12.2022 р.)
за поданням Вченої ради інженерно-хімічного факультету (протокол №9 від 26.09.2022 р.)*

Електронне мережне навчальне видання

ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. МЕТОДОЛОГІЇ ТА ЗАСОБИ РОЗРОБКИ КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Укладач: *Ковалюк Дмитро Олександрович, канд. техн. наук, доц.*

Відповідальний
редактор *Цапар Віталій Степанович, канд. техн. наук, доцент.*

Рецензенти: *Гулієнко Сергій Валерійович., канд. техн. наук, доцент*

Посібник призначений для формування у студентів комплексу знань, умінь та досвіду, необхідних для створення програмного забезпечення комп'ютерних систем керування та інформаційних систем. Дозволяє скорочено за формою, але всебічно за змістом викласти студентам технології розробки та функціонування програмного забезпечення розподілених систем.

© КПІ ім. Ігоря Сікорського, 2022

ВСТУП

Однією з проблем, що виникає в даний час перед спеціалістами, які займаються проектуванням та створенням автоматизованих систем керування, є ефективне використання сучасних засобів розробки програмного забезпечення.

В умовах зростаючої складності програмного забезпечення необхідним є вивчення сучасних засобів розробки, тестування та розгортання, принципів написання гарного коду.

Під час виконання робіт студенти оволодіють моделями розробки програмного забезпечення (зокрема гнучкою методологією Agile, та її реалізацією SCRUM, KANBAN), програмними засобами управління проектами (Jira) системою контролю версій (Git) принципами розробки (DRY, KISS, SOLID), патернами проектування (поведінкові, структурні, породжувальні), методами тестування програмного забезпечення (unit – фреймворки), процесом CI/CD для розгортання та публікації.

ЗМІСТ

Вступ.....	3
Практична робота 1	
Система управління проектами – Jira.....	12
Практична робота 2	
Основи роботи з Git.....	14
Практична робота 3	
Основи галуження, Git на сервері.....	28
Практична робота 4	
Патерни проектування.....	39
Практична робота 5	
Модульне тестування	40
Практична робота 6	
Інтеграційне тестування.....	46
Практична робота 7	
GitLab CI/CD.....	50
Літературні джерела.....	55

Практична робота №1

Система управління проектами - Jira

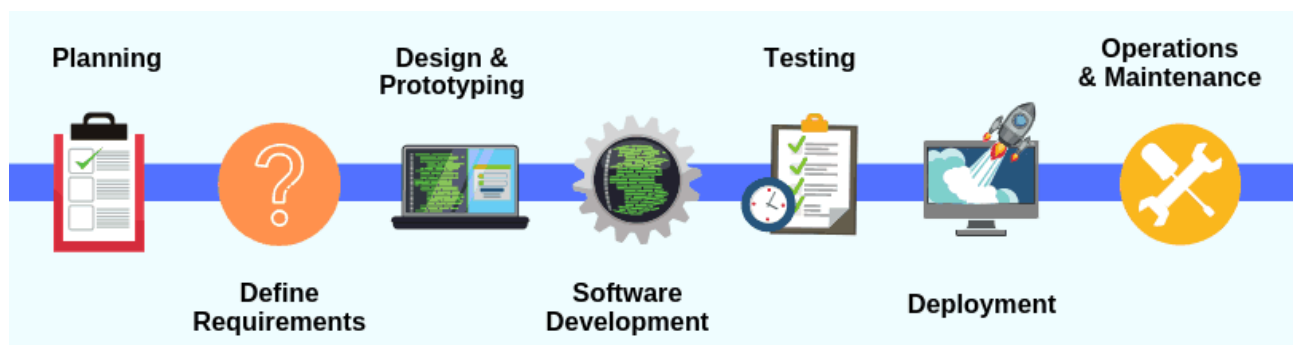
Мета роботи – набути новичок командної роботи з системами управління проектами.

Теоретичні відомості.

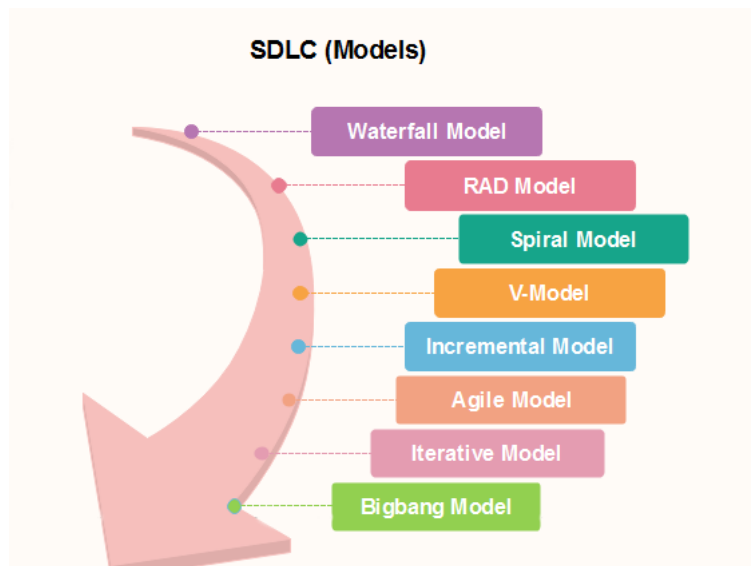
Сьогодні розробка програмного забезпечення проводиться за деякою моделлю чи методологією, яка реалізує етапи життєвого циклу.

До основних етапів життєвого циклу відносять:

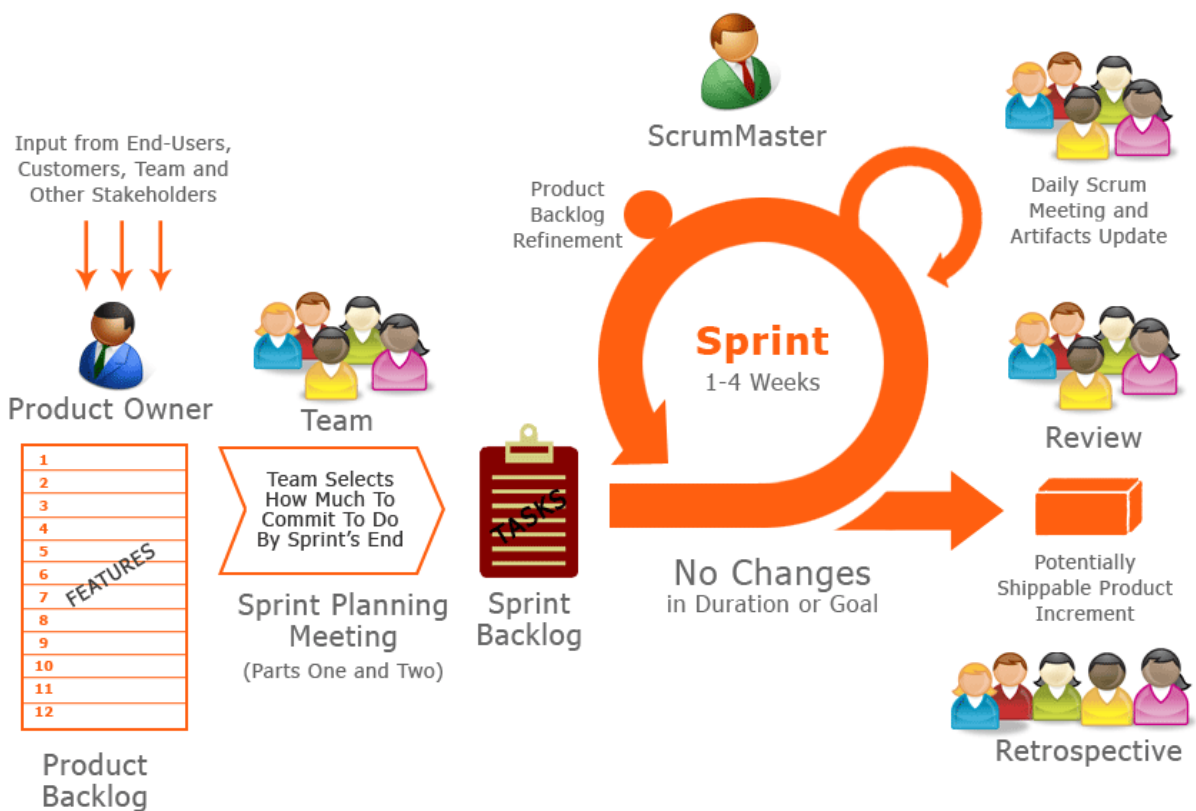
- Planning (Планування і аналіз потреб)
- Analysis (Аналіз вимог)
- Design (Проектування)
- Development (Розробка)
- Testing (Тестування)
- Deployment (Розгортання)
- Maintenance (Підтримка)

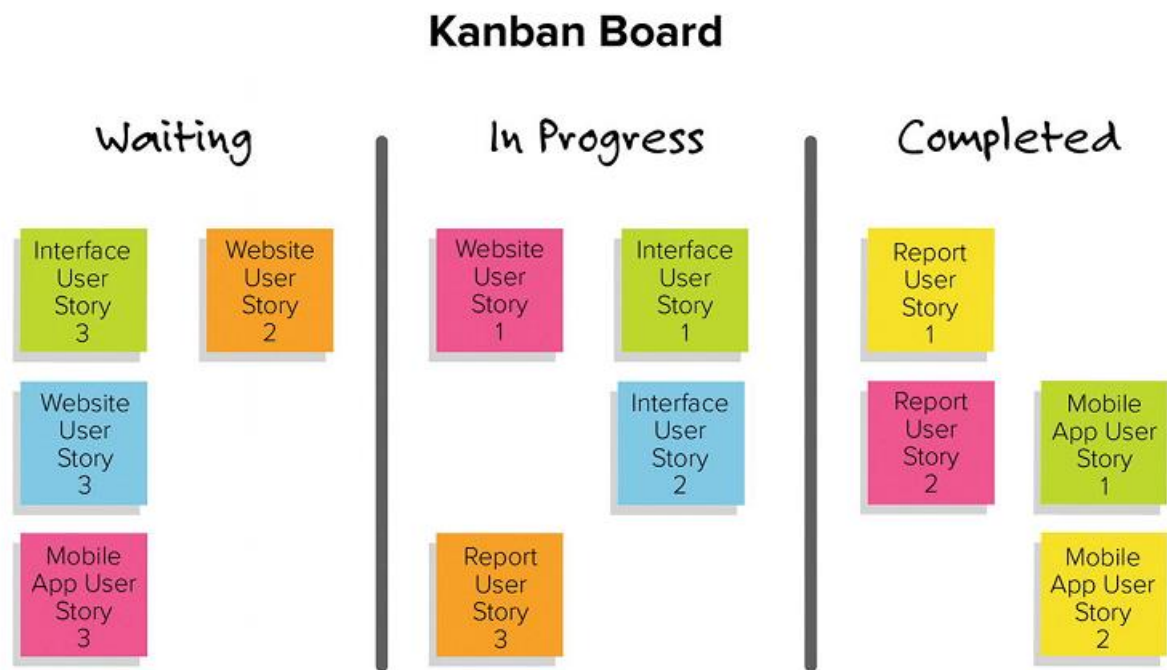


Етапи життєвого циклу пов'язані між собою в моделі життєвого циклу. В деяких моделях етапи повторюються, в інших, ні. Найбільш популярні моделі життєвого циклу наведено на рисунку.



На практиці для багатьох проектів використовуються «гнучкі» Agile-методології SCRUM та Kanban. Ідеї обох підходів показано на рисунках





Для зручної імплементації методологій розробки використовуються системи управління проектами, які дозволяють створювати задачі, закріплювати їх за виконавцями, моніторити статус виконання та ін.

Однією з найбільш поширених систем є Jira від Atlassian <https://www.atlassian.com/software/jira>

Вона передбачає як платний, так і безкоштовний функціонал і дозволяє реалізувати SCRUM і Kanban.

Хід роботи

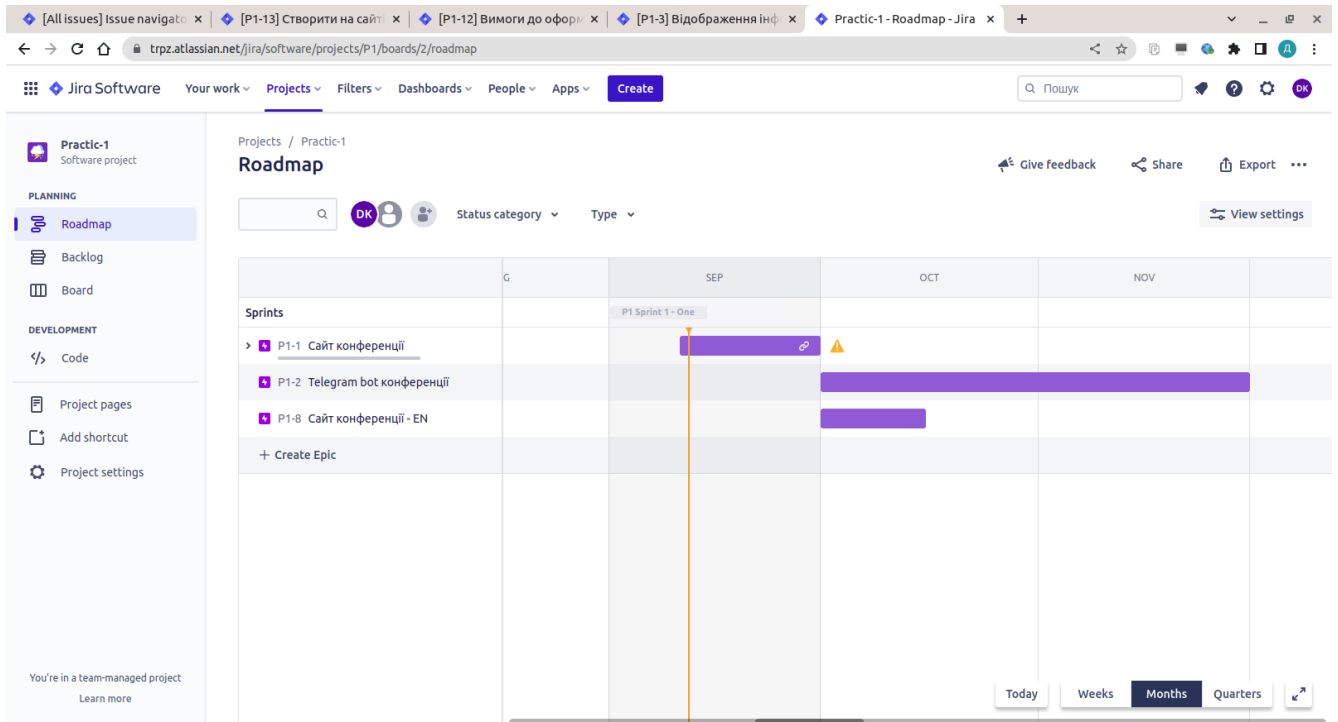
1. Проаналізувати існуючі системи управління проектами.
2. Вибрати систему управління проектом. Рекомендовано Jira від Atlassian.
3. Створити 2 проекти для методологій SCRUM та Kanban.

SCRUM	Kanban
Додати учасників проекту (членів бригади)	
Створити дорожню карту Roadmap – повинна містити мінімум 2 «епіки». Один з «епіків» має включати «історії» та «задачі»	
Сформувані перелік задач для розробки ПЗ (backlog). Наприклад: • Зареєструвати домен • Розробити дизайн • Спроектувати базу даних • Спроектувати архітектуру сайту • Зареєструвати хостинг Мінімальна кількість задач – 10. Згрупувати задачі по історіям. Мінімальна кількість історій – 2	
Вибрати задачі в беклог спринта	Задати такі статуси обробки задач: • SELECTED FOR DEVELOPMENT • IN PROGRESS • TESTING • DONE
Розподілити задачі між учасниками проекту	
Оцінити задач учасниками проекту	
Стартувати спринт	
Закрити задачі, змінюючи статус їх виконання, перевірити показники	
Закрити спринт	
Створити dashboard, на якому розмістити віджети, що відображають статистику роботи. Кількість і перелік віджетів визначається членами бригади і залежить від можливостей обраної системи.	

Приклад виконання

Розробити програмне забезпечення для проведення міжнародної наукової конференції.

Створюємо Roadmap, епіки, початок та кінець виконання



Створюємо stories

The screenshot shows the Jira 'All issues' view for project 'Practic-1'. The left sidebar contains navigation options: Filters (Search issues, OTHER, My open issues, Reported by me, All issues, Open issues, Done issues, Viewed recently, Created recently, Resolved recently, Updated recently, View all filters). The main area displays the 'All issues' view with a table of issues. The table columns include Key, Summary, Assignee, Reporter, P, Status, Resolution, Created, Updated, and Due.

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	P1-15	Вибір шаблону сайту	Дмитро Ковалюк	Дмитро Ковалюк	==	TO DO	Unresolved	11/веп./22	11/веп./22	
	P1-5	Розгортання і публікація сайту	Дмитро Ковалюк	Дмитро Ковалюк	==	TO DO	Unresolved	11/веп./22	12/веп./22	
	P1-4	Реєстрація і подача матеріалів	Unassigned	Дмитро Ковалюк	==	TO DO	Unresolved	11/веп./22	11/веп./22	...
	P1-3	Відображення інформації про конференцію	Unassigned	Дмитро Ковалюк	==	TO DO	Unresolved	11/веп./22	11/веп./22	

Створюємо tasks

The screenshot shows a Jira issue page for the issue titled "Реєстрація субдомену" (Subdomain Registration). The issue is in the "To Do" status. The description states: "В конструкторському бюро КБ ІС зареєструвати домен <https://akit.kpi.ua/>". The issue is assigned to "Дмитро Ковалюк" (Dmitro Kovalyuk). The "Pinned fields" section shows the Assignee, Labels, Sprint, Story point estimate, and Reporter. The "Activity" section shows a comment by "DK" (Dmitro Kovalyuk) with the text "Add a comment...". The "Linked issues" section shows a link to "P1-5 Розгортання і публікація сайту".

Filters

Search issues

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

View all filters

Реєстрація субдомену

Attach Add a child issue Link issue

Description

В конструкторському бюро КБ ІС зареєструвати домен <https://akit.kpi.ua/>

Linked issues

blocks

P1-5 Розгортання і публікація сайту

Activity

Show: All Comments History

Newest first

DK Add a comment...

Pro tip: press **M** to comment

To Do

Pinned fields

Click on the next to a field label to start pinning.

Details

Assignee DK Дмитро Ковалюк

Labels None

Sprint None +1

Story point estimate 4

Reporter DK Дмитро Ковалюк

Created 11 годин тому

Updated 11 хвилин тому

Configure

The screenshot shows the "All issues" page in Jira. The page displays a list of 8 issues, all assigned to "Дмитро Ковалюк" (Dmitro Kovalyuk) and in the "To Do" status. The issues are related to the registration and deployment of a website. The table below shows the details of these issues.

Filters

Search issues

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

View all filters

All issues — Edited Save as

Project: All Task Status: All Assignee: All + More Contains text Search Switch to JQL

1-8 of 8

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
☒	P1-16	Пошук адаптивних шаблонів	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	...
☒	P1-14	Створити Google форму для подачі матеріалів	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-13	Створити на сайті ФЗЗ для подачі матеріалів	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-12	Вимоги до оформлення тез доповіді	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-11	Розділ "Тематики конференції"	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-10	Розділ "Організатори конференції"	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-7	Реєстрація електронної пошти	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	
☒	P1-6	Реєстрація субдомену	Дмитро Ковалюк	Дмитро Ковалюк	TO DO	Unresolved		11/веп./22	12/веп./22	

1-8 of 8

Відбираємо задачі для спринта

Projects / Practic-1

Backlog

Search [DK] Epic Type Insights

P1 Sprint 1 - One 2 11 Sep - 25 Sep (5 issues) 49 0 0 **Start sprint**

- ☒ P1-7 Реєстрація електронної пошти **САЙТ КОНФЕРЕНЦІЇ** 4 TO DO DK
- ☒ P1-6 Реєстрація субдомену **САЙТ КОНФЕРЕНЦІЇ** 5 TO DO DK
- ☒ P1-10 Розділ "Організатори конференції" **САЙТ КОНФЕРЕНЦІЇ** 10 TO DO DK
- ☒ P1-11 Розділ "Тематики конференції" **САЙТ КОНФЕРЕНЦІЇ** 10 TO DO DK
- ☒ P1-16 Пошук адаптивних шаблонів **САЙТ КОНФЕРЕНЦІЇ** 20 TO DO DK

+ Create issue

Backlog (7 issues) 45 0 0 **Create sprint**

- ☒ P1-13 Створити на сайті ФЗЗ для подачі матеріалів **САЙТ КОНФЕРЕНЦІЇ** 20 TO DO DK
- ☒ P1-3 Відображення інформації про конференцію **САЙТ КОНФЕРЕНЦІЇ** TO DO
- ☒ P1-4 Реєстрація і подача матеріалів **САЙТ КОНФЕРЕНЦІЇ** TO DO
- ☒ P1-5 Розгортання і публікація сайту **САЙТ КОНФЕРЕНЦІЇ** TO DO DK
- ☒ P1-12 Вимоги до оформлення тез доповідей **САЙТ КОНФЕРЕНЦІЇ** 15 TO DO DK

You're in a team-managed project. Learn more

Projects / Practic-1

P1 Sprint 1

9 days remaining **Complete sprint**

GROUP BY None Insights

TO DO 5 ISSUES

- Реєстрація електронної пошти **САЙТ КОНФЕРЕНЦІЇ**
 - ☒ P1-7 4 DK
- Реєстрація субдомену **САЙТ КОНФЕРЕНЦІЇ**
 - ☒ P1-6 5 DK
- Розділ "Організатори конференції" **САЙТ КОНФЕРЕНЦІЇ**
 - ☒ P1-10 10 DK

IN PROGRESS

Є ПИТАННЯ

DONE ✓

Sprint started
We've taken you to your team's sprint board. Good luck team!

You're in a team-managed project. Learn more

Створюємо дошку із статусами виконання задач

Projects / Practic-1

P1 Sprint 1

9 days remaining Complete sprint

GROUP BY Subtask Insights

TO DO 1 ISSUE

IN PROGRESS 2 ISSUES

Є ПИТАННЯ 1 ISSUE

DONE 1 ISSUE ✓

Everything else 5 issues

Розділ "Тематики конференції" САЙТ КОНФЕРЕНЦІЇ

✓ P1-11 10 DK

Пошук адаптивних шаблонів САЙТ КОНФЕРЕНЦІЇ

✓ P1-16 20 DK

Розділ "Організатори конференції" САЙТ КОНФЕРЕНЦІЇ

✓ P1-10 10 DK

Регістрація субдомену САЙТ КОНФЕРЕНЦІЇ

✓ P1-6 5 DK

Регістрація електронної пошти САЙТ КОНФЕРЕНЦІЇ

✓ P1-7 4 DK

You're in a team-managed project. Learn more

Закриваємо деякі задачі, перевіряємо показники ефективності

Projects / Practic-1

P1 Sprint 1

9 days remaining Complete sprint

GROUP BY Subtask Insights

TO DO 1 ISSUE

IN PROGRESS 2 ISSUES

Є ПИТАННЯ 1 ISSUE

DONE 1 ISSUE ✓

Everything else 5 issues

Розділ "Тематики конференції" САЙТ КОНФЕРЕНЦІЇ

✓ P1-11 10 DK

Пошук адаптивних шаблонів САЙТ КОНФЕРЕНЦІЇ

✓ P1-16 20 DK

Розділ "Організатори конференції" САЙТ КОНФЕРЕНЦІЇ

✓ P1-10 10 DK

Регістрація субдомену САЙТ КОНФЕРЕНЦІЇ

✓ P1-6 5 DK

Регістрація електронної пошти САЙТ КОНФЕРЕНЦІЇ

✓ P1-7 4 DK

Insights P1 SPRINT 1

Sprint progress

9% done

Done In progress Not started

9% 71% 20%

Sprint burndown

4 points done, 45 points to go

On track

100% 80% 60% 40% 20% 0%

Sep 12 Sep 25

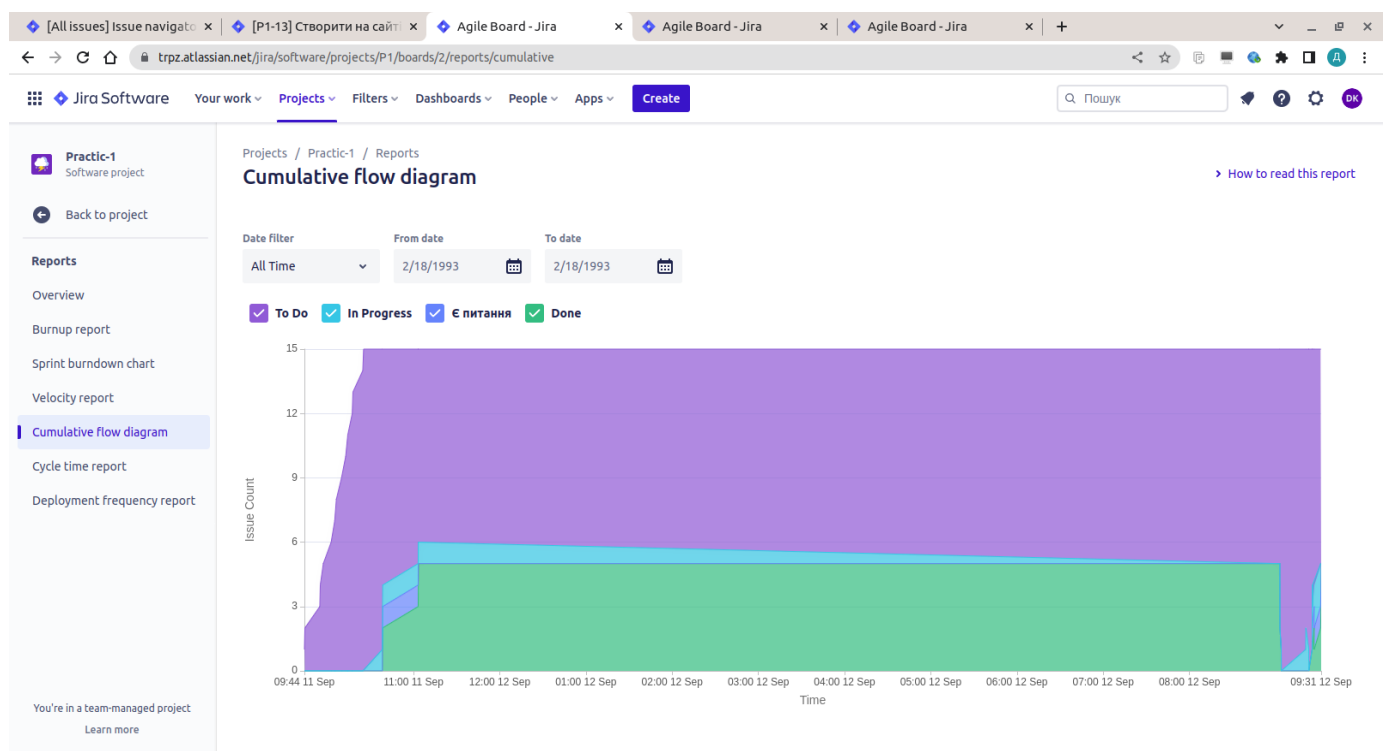
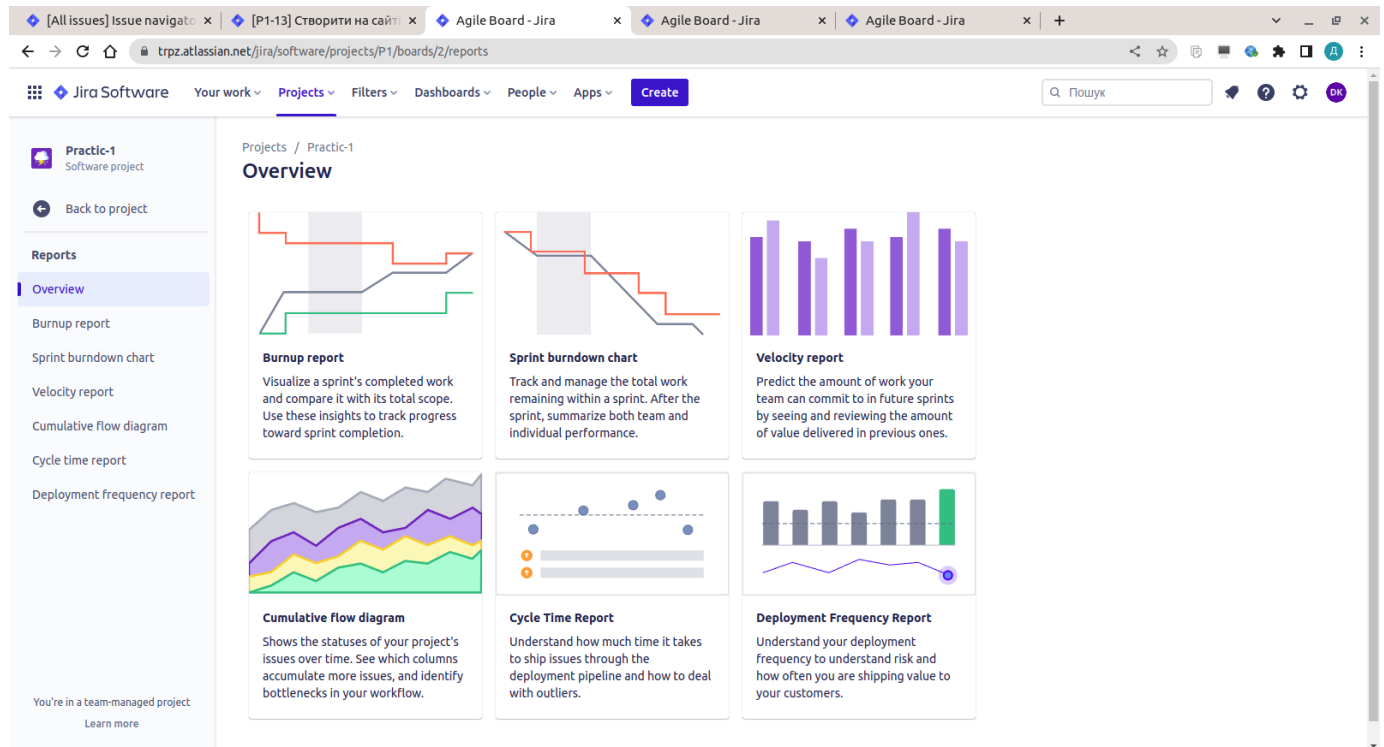
Remaining work Guideline

Epic progress

This sprint is working towards 1 epic

P1-1 Сайт конференції 8% done

Виконуємо всі задачі, закриваємо спринт. Дивимось звіти



Контрольні питання

1. Життєвий цикл (SDLC).
2. SCRUM.
3. Основні функції систем управління проектами.
4. Метод критичного шляху.

Практична робота №2

Основи роботи з Git

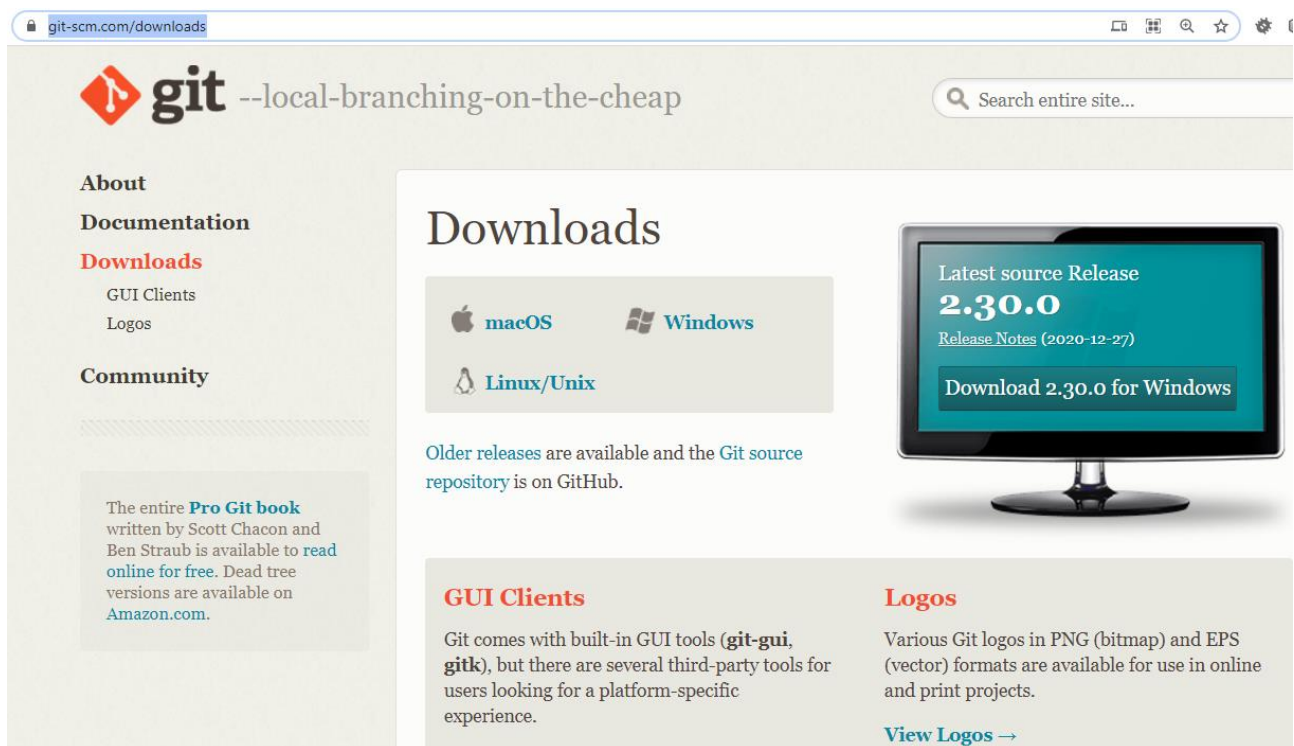
Мета роботи – встановити на ПК систему контролю версій Git, опанувати основні команди для роботи в локальній копії.

Хід роботи

Частина 1 - Встановлення

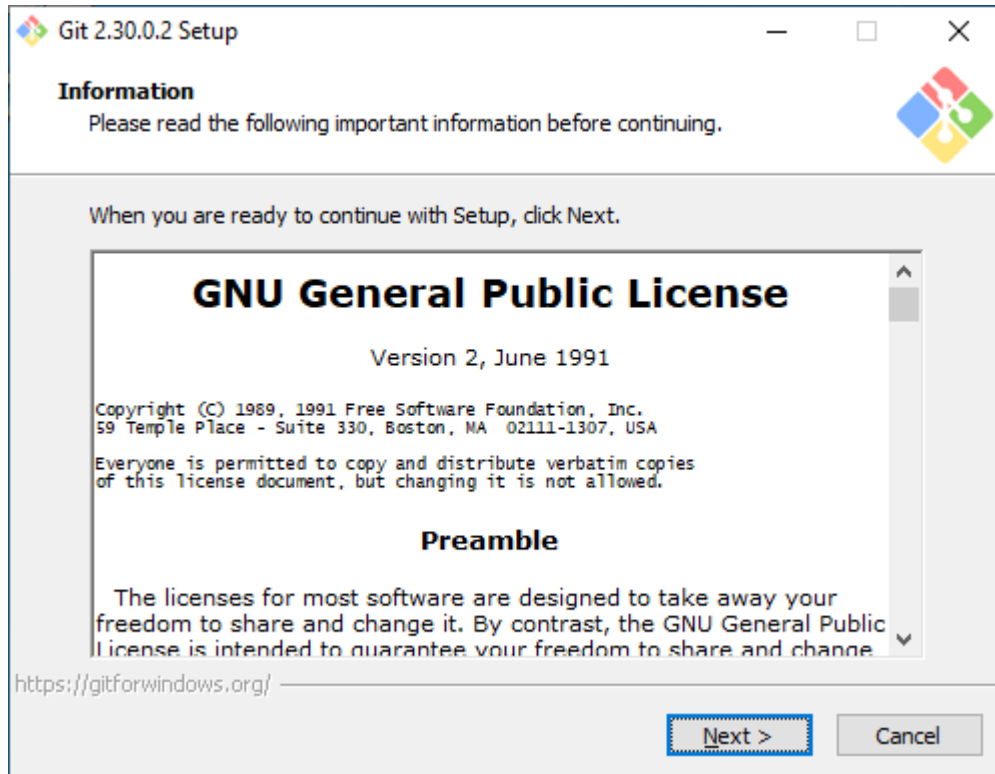
1. Зайти на офіційний сайт і перейти на сторінку завантаження

<https://git-scm.com/downloads>

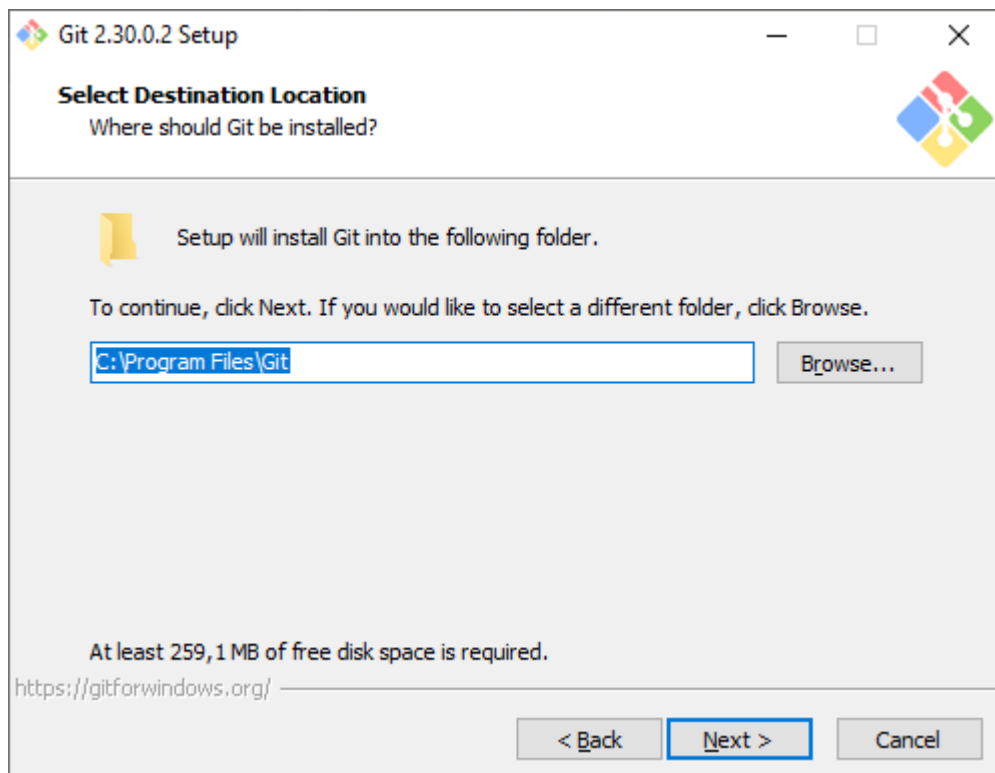


2. Вибрати завантаження для відповідної операційної системи. Далі описується процес встановлення для Windows. Зазначимо, що доцільно притримуватися (вибирати ті опції), які пропонує інсталятор.

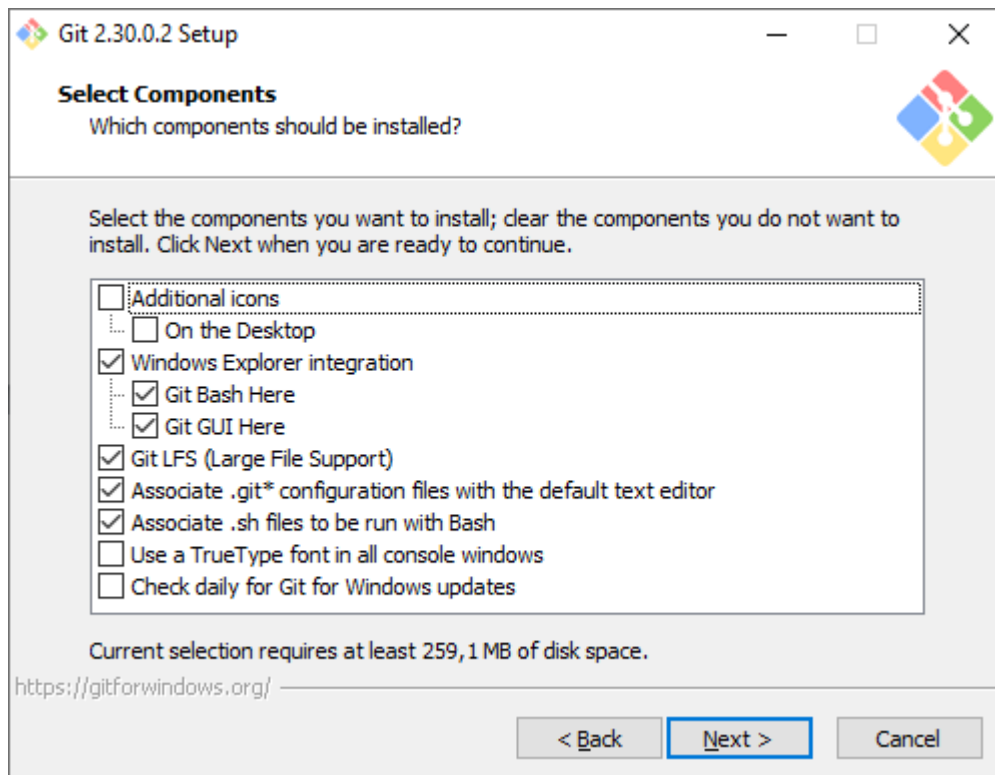
Запустити інсталятор



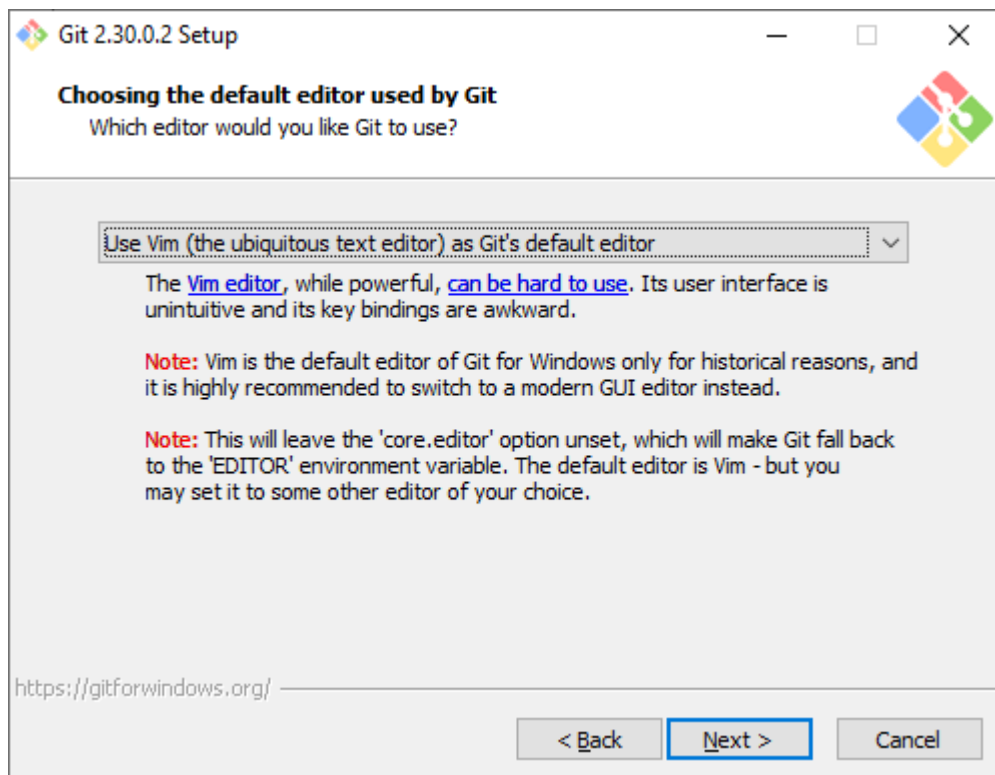
Вибрати директорію встановлення



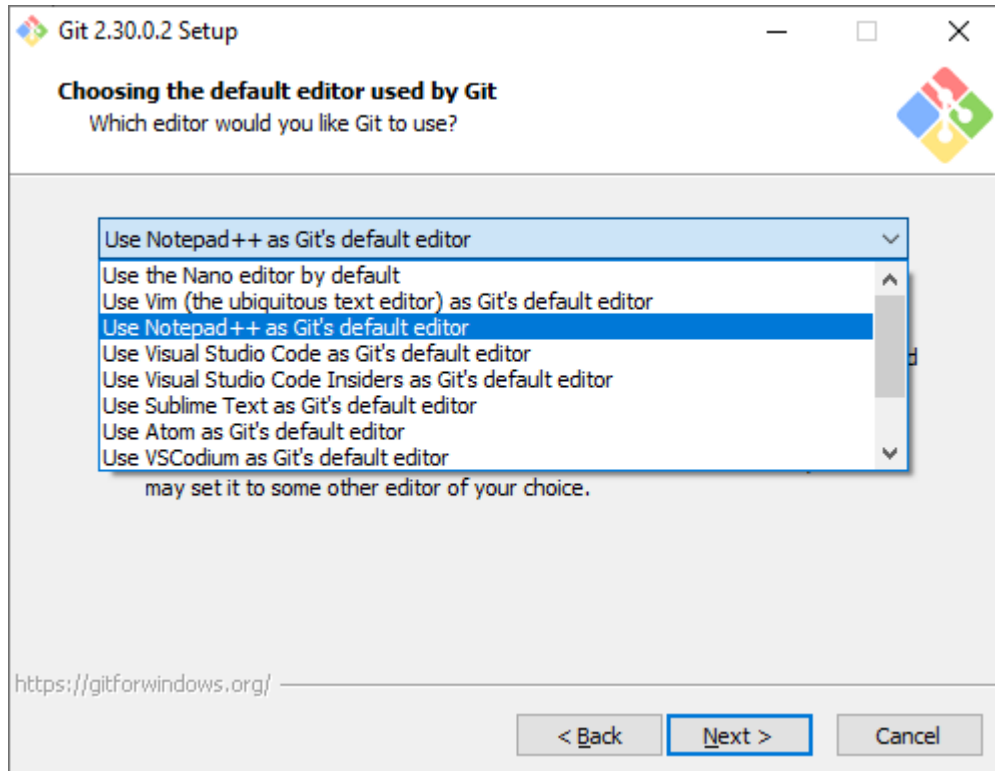
Вибрати компоненти для встановлення (залишаємо значення за замовчуванням).
В цьому випадку поставляться додаткові пункти контекстного меню програми «Провідник Windows»



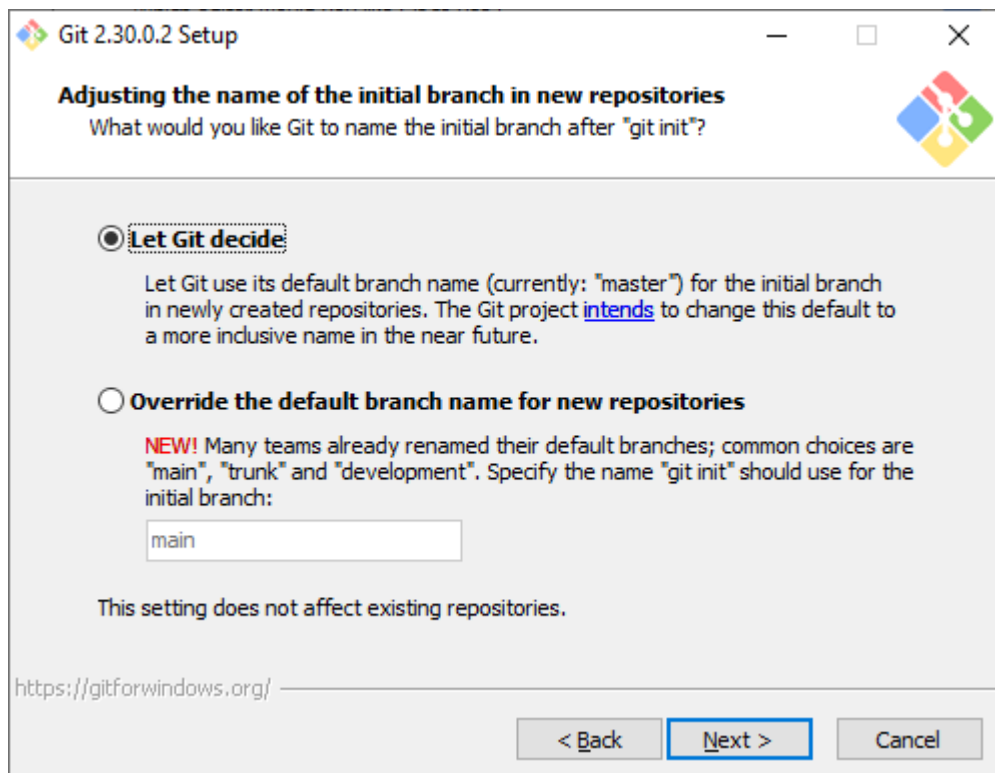
Вказуємо текстовий редактор, який буде запускатися для введення коментарів.



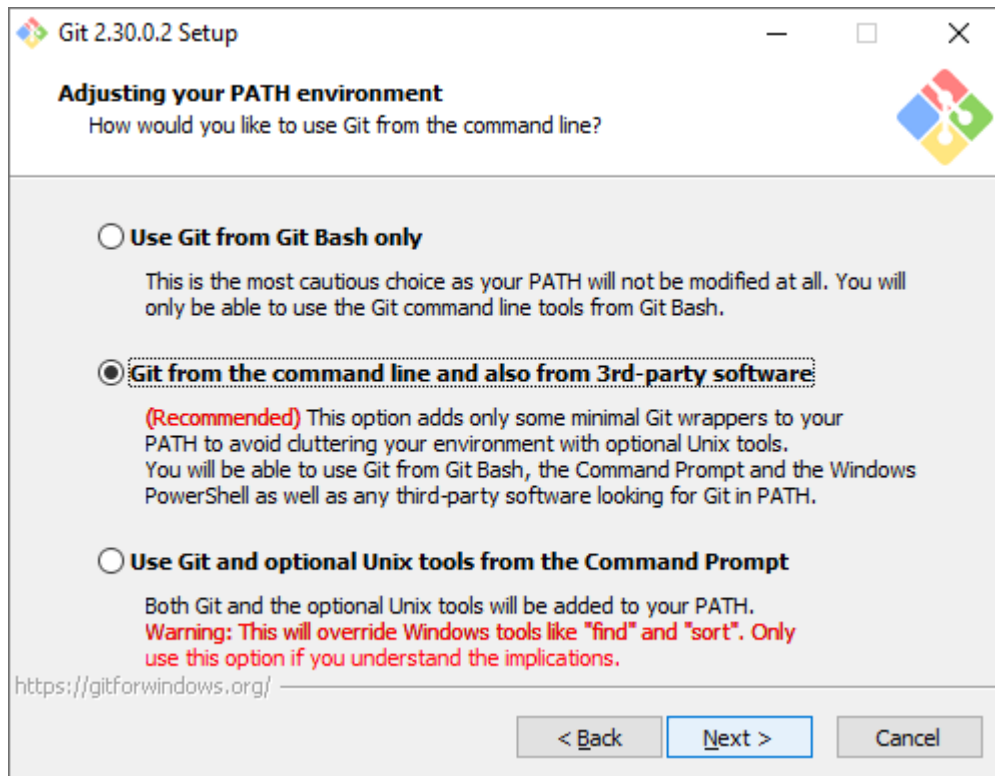
Рекомендується вибрати якийсь редактор з графічним інтерфейсом



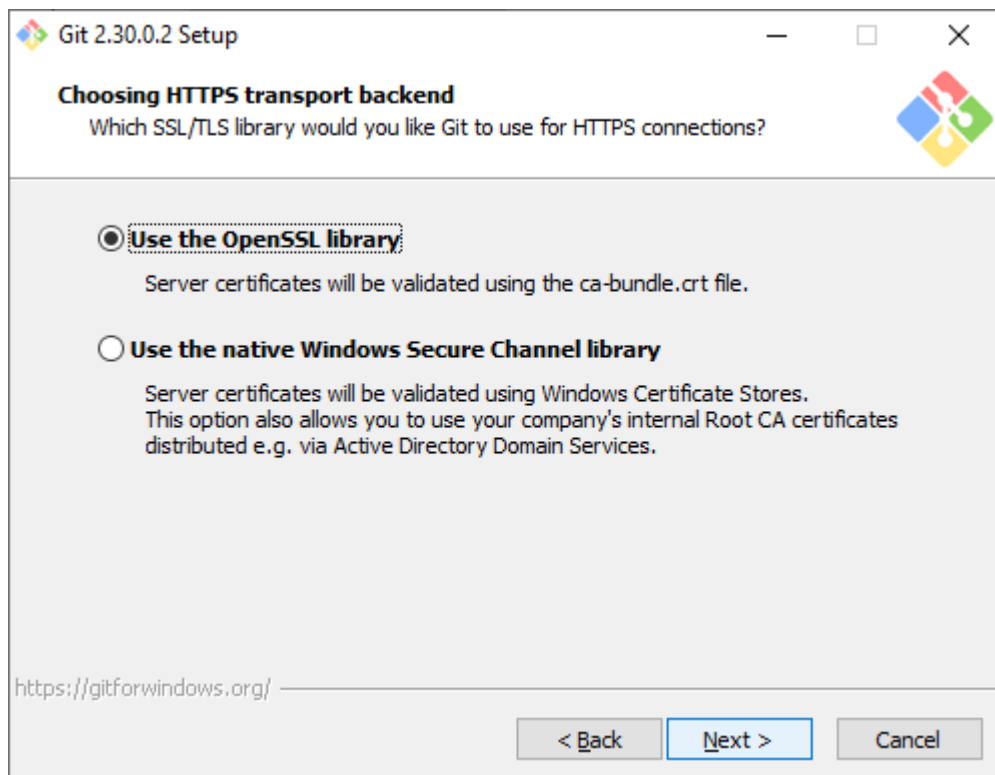
Вибираємо назву гілки за замовчуванням (залишаємо «master»)



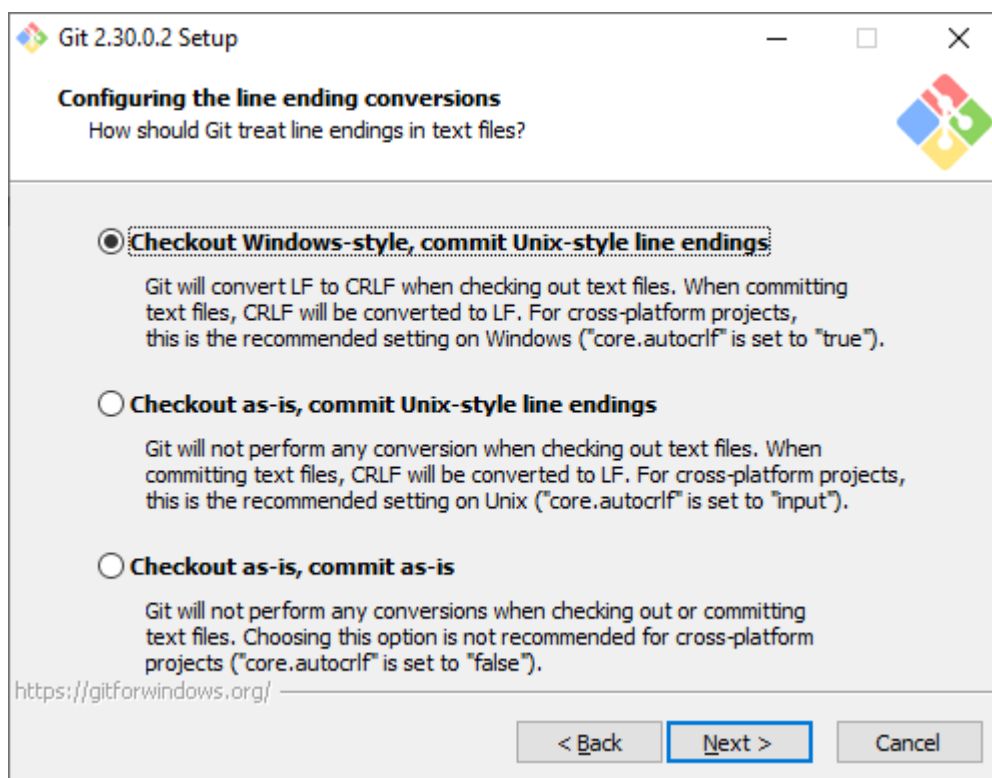
Дозволяємо запускати Git як з командного рядка, так і з додатків інших розробників. (наприклад можна користуватися Git з консолі OpenServer)



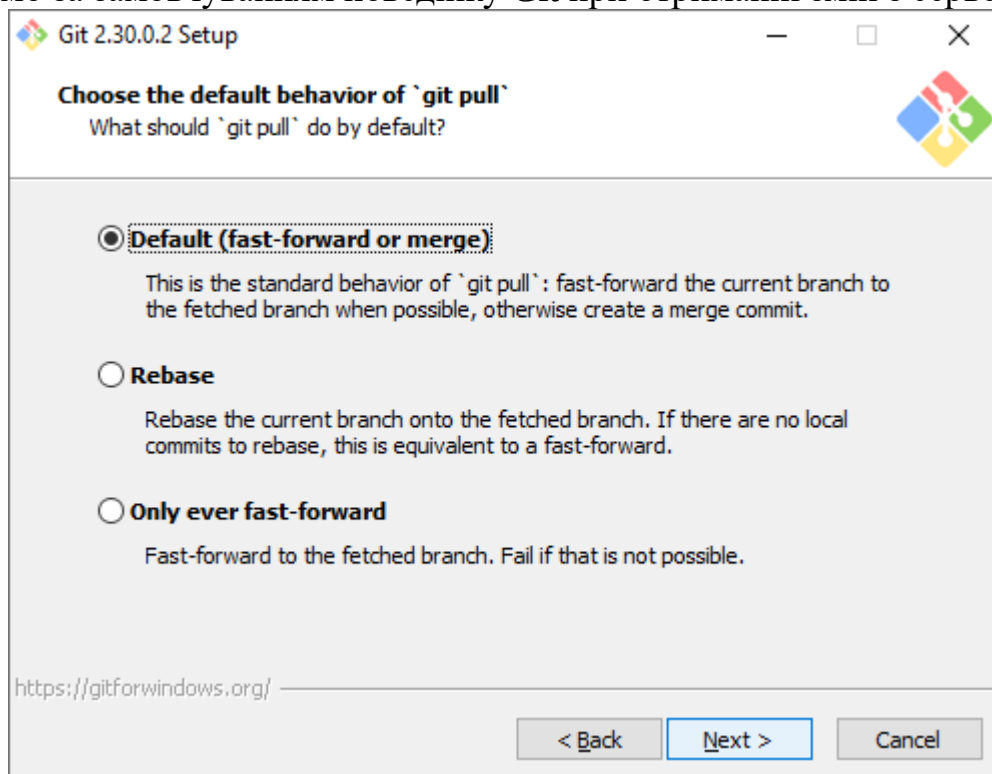
Вибираємо тип перевірки сертифікату серверу, при передачі даних через Інтернет по https протоколу



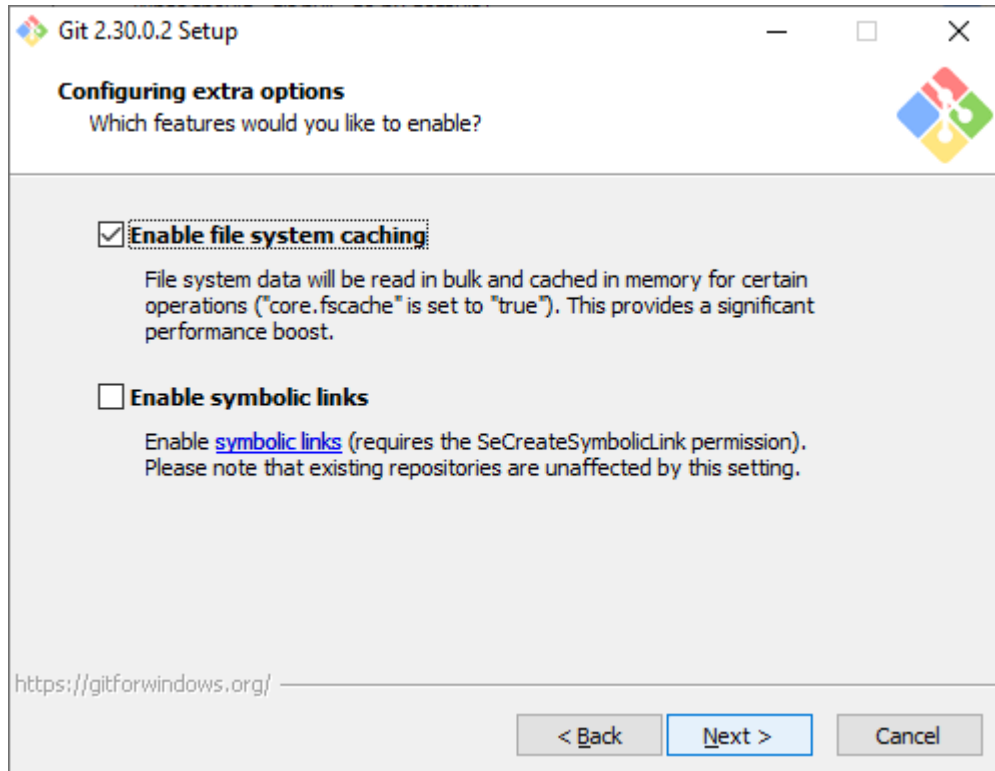
Дозволяємо Git – виконувати конвертування символів закінчення строки, для уникнення проблем, якщо розробники працюють з різних ОС.



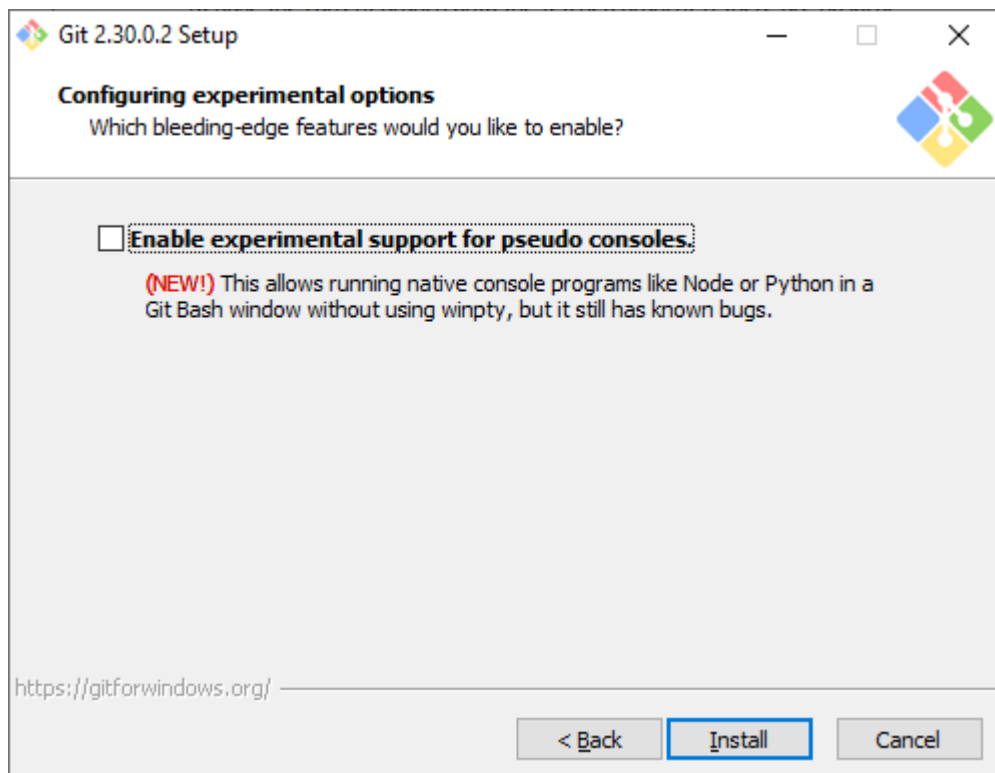
Залишаємо за замовчуванням поведінку Git при отриманні змін з серверу



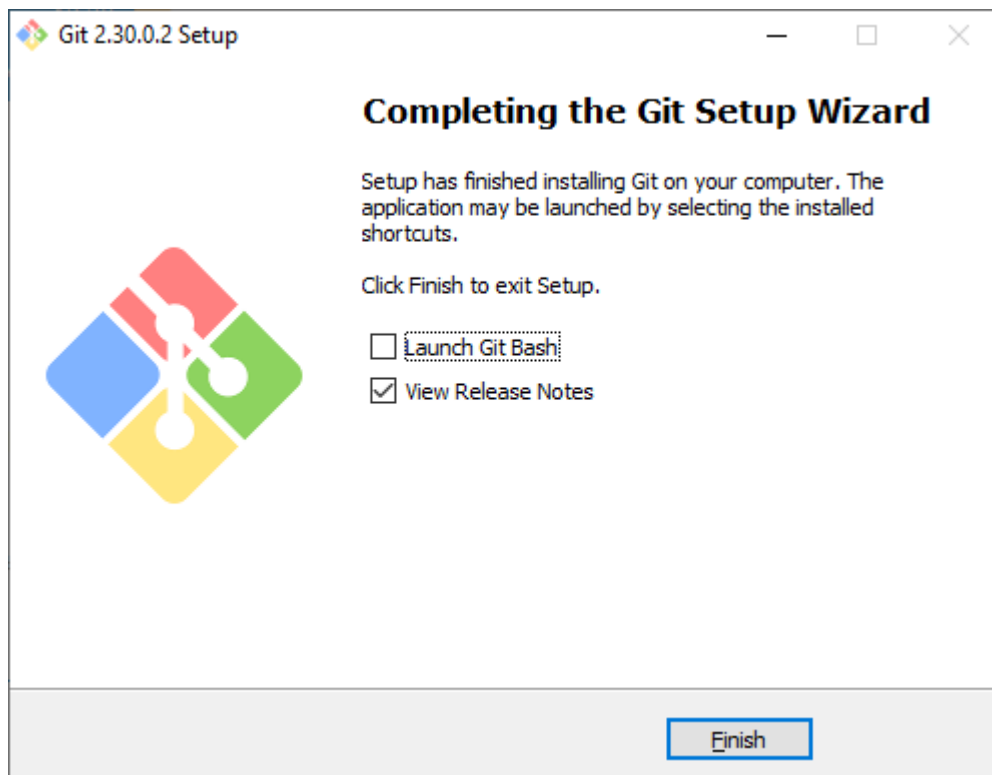
Дозволяємо кешування для збільшення швидкодії



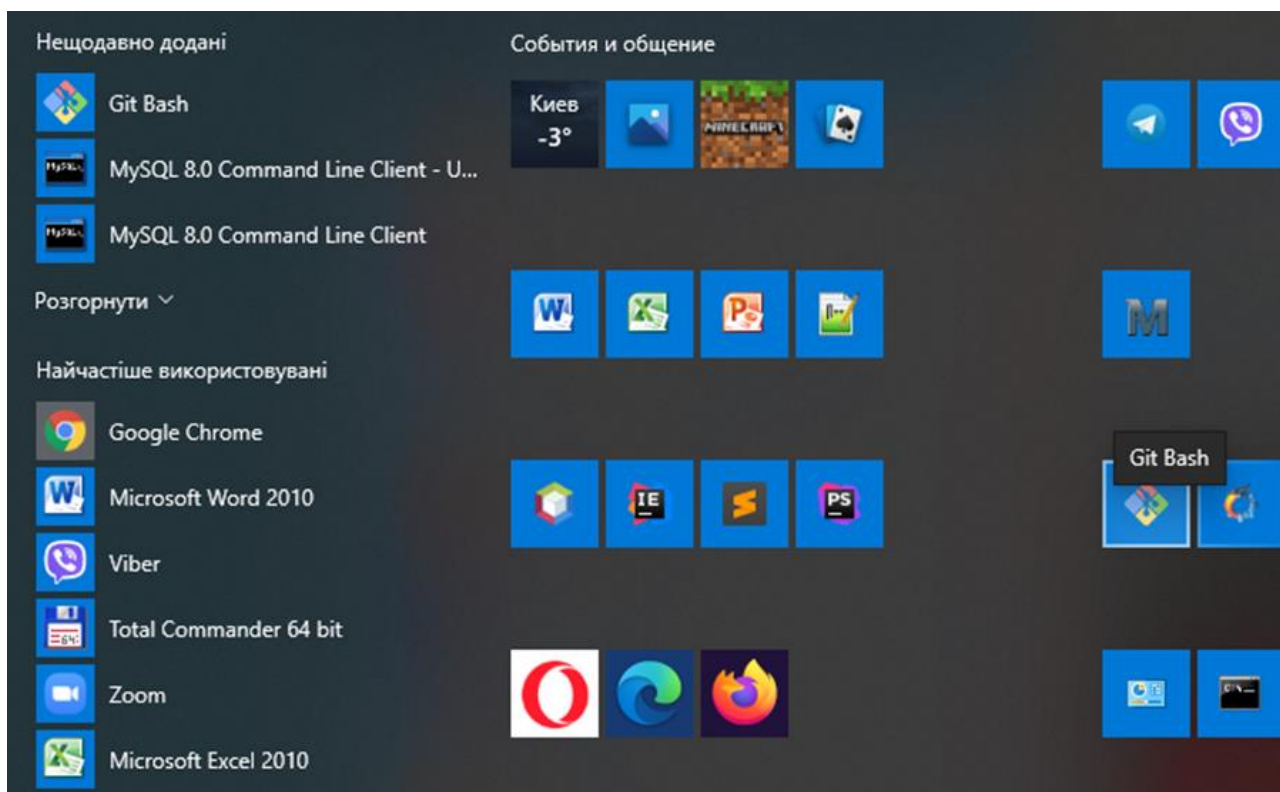
Остання галочка на розсуд студентів.



Результат успішного встановлення.



Запуск Git з консолі (cmd) та через Git Bash



```
cmd. Командний рядок
Microsoft Windows [Version 10.0.19041.746]
(c) Корпорація Майкрософт (Microsoft Corporation), 2020. Усі права захищені.

C:\Users\Dima>git --version
git version 2.30.0.windows.2

C:\Users\Dima>

MINGW64:/c
Dima@DESKTOP-3BTJIEQ MINGW64 ~
$ cd ../

Dima@DESKTOP-3BTJIEQ MINGW64 /c/Users
$ cd ../

Dima@DESKTOP-3BTJIEQ MINGW64 /c
$ git --version
git version 2.30.0.windows.2

Dima@DESKTOP-3BTJIEQ MINGW64 /c
$ |
```

Створюємо репозиторій.

```
Dima@DESKTOP-3BTJIEQ MINGW64 /d
$ cd git/ppz/lections/

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/ppz/lections
$

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/ppz/lections
$ git init
Initialized empty Git repository in D:/git/ppz/lections/.git/

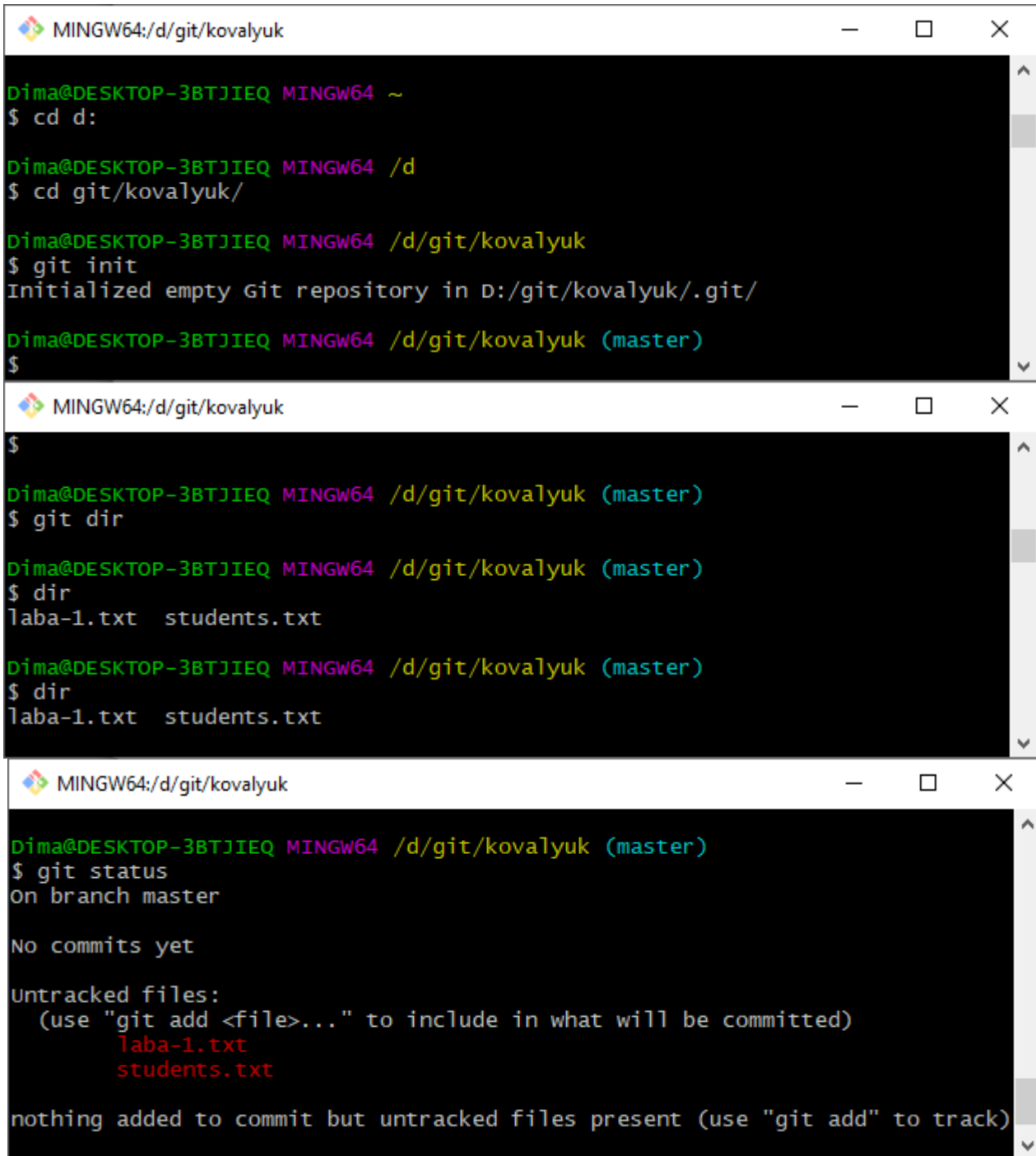
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/ppz/lections (master)
$ |
```

Частина 2. Основні команди

Хід виконання

Що потрібно зробити	команда Git
Створити репозиторій в папці «Прізвище студента» - наприклад, ivanov	git init
Задати своє ім'я та електронну пошту	git config
Створити в директорії репозиторія 2 файли: - students.txt (містить довільні 3 прізвища) - laba-1.txt (містить довільну строку)	
Перевірити статус цих файлів	git status
Додати файли під версійний контроль	git add git status git commit
Перевірити статус файлів	git status
В файлі students.txt – додати ще два нових прізвища	
Перевірити статус файлів	git status
Перевірити зміни в файлах в порівнянні з попереднім збереженням	git diff
Зберегти змінений файл	git add git commit
В файлі students.txt – додати ще два нових прізвища	
Перевірити статус файлів	git status
Додати файл в індекс	git add
Перевірити зміни в файлах	git diff
Видалити файл з індексу	git reset
Відмінити зміни в файлі, зроблені після останнього збереження	git checkout
Видалити файл laba-1.txt з під версійного контролю	git rm
Перевірити статус файлів	git status
Створити файл .gitignore	
В.gitignore вказати, що файли з розширенням ini та папки IDE та vendors не будуть відслідковуватися	
Зберегти файл .gitignore	git add git commit
Створити в робочій директорії папки IDE та vendors, а також файл application.ini	
Перевірити статус файлів	

Приклад виконання



```
MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 ~
$ cd d:

Dima@DESKTOP-3BTJIEQ MINGW64 /d
$ cd git/kovalyuk/

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk
$ git init
Initialized empty Git repository in D:/git/kovalyuk/.git/

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$

MINGW64:/d/git/kovalyuk

$

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git dir

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ dir
laba-1.txt  students.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ dir
laba-1.txt  students.txt

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        laba-1.txt
        students.txt

nothing added to commit but untracked files present (use "git add" to track)
```



```
MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git add *

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   laba-1.txt
        new file:   students.txt

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git commit -m 'first edition'
[master (root-commit) 179077f] first edition
 2 files changed, 6 insertions(+)
 create mode 100644 laba-1.txt
 create mode 100644 students.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ |

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
On branch master
nothing to commit, working tree clean

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ |

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
On branch master
changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   students.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git diff
diff --git a/students.txt b/students.txt
index 6bb39ae..914a8fe 100644
--- a/students.txt
+++ b/students.txt
@@ -1,4 +1,4 @@
  Ковалюк
-Ковалюк
  Сазонов
-Онїщенко
\ No newline at end of file
+Онїщенко
+Цапар

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git commit -a -m 'add new students'
[master c62499a] add new students
1 file changed, 2 insertions(+), 2 deletions(-)

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git diff
diff --git a/students.txt b/students.txt
index 914a8fe..6f71c7f 100644
--- a/students.txt
+++ b/students.txt
@@ -1,4 +1,6 @@
+Лукїнюк
  Ковалюк
+Ярошук
  Сазонов
  Онїщенко
-Цапар
+Цапар
\ No newline at end of file
```

```
MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git reset HEAD students.txt
Unstaged changes after reset:
M      students.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
on branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   students.txt

no changes added to commit (use "git add" and/or "git commit -a")

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git diff
diff --git a/students.txt b/students.txt
index 6f71c7f..e2fbfcb 100644
--- a/students.txt
+++ b/students.txt
@@ -1,6 +1,6 @@
-Лукінюк
  Ковалюк
  Ярошук
+Лукінюк
  Сазонов
  Оніщенко
  Цапар
\ No newline at end of file

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$

MINGW64:/d/git/kovalyuk

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git checkout students.txt
Updated 1 path from the index

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
on branch master
nothing to commit, working tree clean

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$
```

```
MINGW64:/d/git/kovalyuk
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git rm laba-1.txt
rm 'laba-1.txt'

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
on branch master
changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   laba-1.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git commit -m 'delete file laba-1.txt'
[master ea719b1] delete file laba-1.txt
1 file changed, 2 deletions(-)
delete mode 100644 laba-1.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ |

MINGW64:/d/git/kovalyuk
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
on branch master
untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git add .gitignore

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git commit -m 'add gitignore file'
[master a9fda52] add gitignore file
1 file changed, 4 insertions(+)
create mode 100644 .gitignore

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ |

MINGW64:/d/git/kovalyuk
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ dir
application.ini  ide  students.txt  vendors

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ git status
on branch master
nothing to commit, working tree clean

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/kovalyuk (master)
$ |
```

Практична робота № 3

Основи галуження, Git на сервері

Мета роботи – засвоїти роботу з локальними та віддаленими гілками в системі контролю версій Git.

Хід роботи

Частина 1 – Робота з локальними гілками

Хід виконання

Що потрібно зробити	команда Git
Створити репозиторій в папці «Прізвище студента» - наприклад, ivanov	git init
Задати текстовий редактор (він буде потрібен для написання повідомлень про злиття)	git config
Створити в директорії репозиторія 2 файли: • students.txt (містить довільні 3 прізвища) • laba-2.txt (містить довільну строку)	
Перевірити статус цих файлів	git status
Додати файли під версійний контроль	add, commit
Створити нову гілку feature/64-login-page	git branch
Переключитися в гілку feature/64-login-page	git checkout
Додати та зберегти файл login.html	add, commit
Повернутися в гілку master	git checkout
Створити та перейти в нову гілку hotfix/68-resize-image	git checkout -b
Додати та зберегти файл img/logo.png	add, commit
Переключитися в гілку master	git checkout
Спробувати видалити гілку hotfix/68-resize-image (git має видати повідомлення про неповне злиття і відмовити)	git branch -d
Влити гілку hotfix/68-resize-image в master	git merge
Видалити гілку hotfix/68-resize-image	git branch -d
Переключитися в гілку feature/64-login-page	git checkout
Додати та зберегти файл login.php	add, commit
Повернутися в гілку master	git checkout
Влити гілку feature/64-login-page в master	git merge
Видалити гілку feature/64-login-page	git branch -d
На кожному кроці переключення між гілками – дивитися на вміст файлів робочої директорії	

Частина 2 – Робота з віддаленими гілками

Хід виконання

1. Зареєструватися на <https://gitlab.com>

GitLab — сайт та система керування репозиторіями програмного коду для Git

GitLab.com

GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab Community Forum](#)
- [GitLab Homepage](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms](#).

Username or email

Password

☐ Remember me [Forgot your password?](#)

Sign in

Don't have an account yet? [Register now](#)

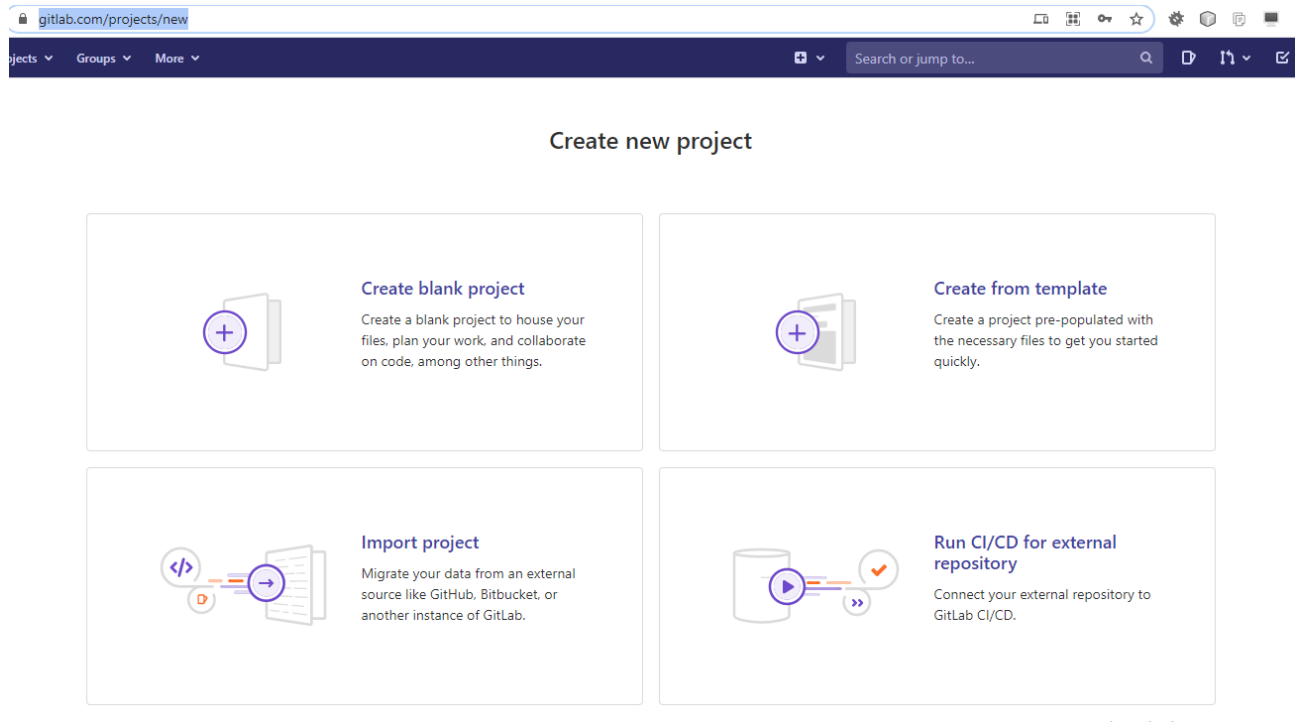
Sign in with

Google GitHub

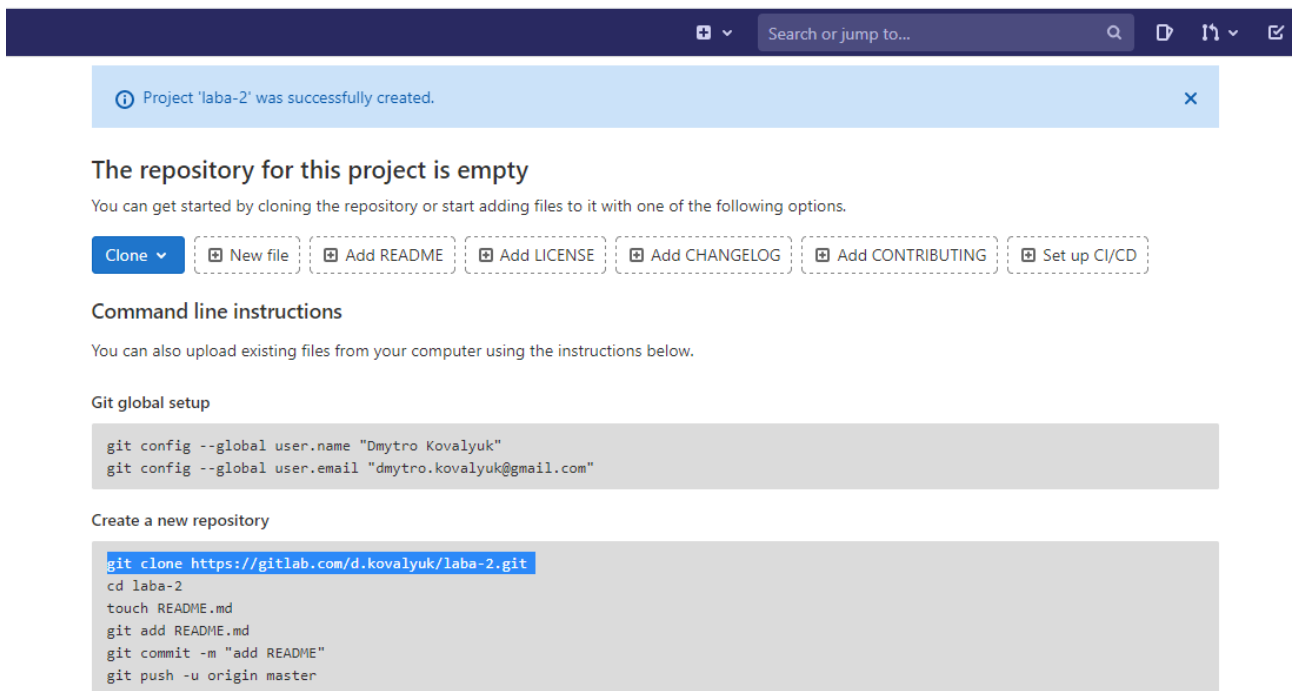
Twitter Bitbucket

2. Створити проект на Gitlab

<https://gitlab.com/projects/new>

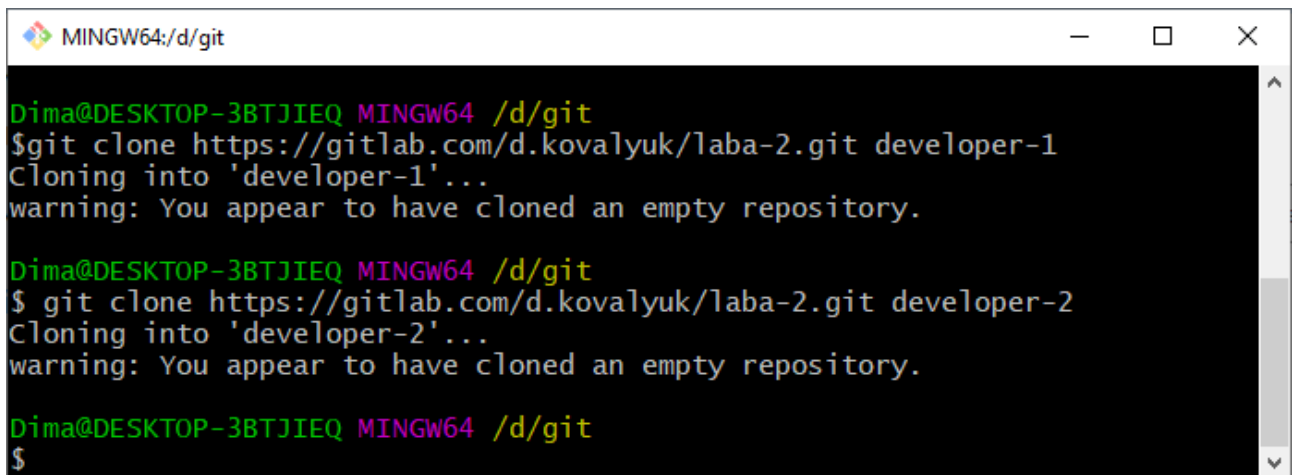
This screenshot shows the 'Create blank project' form on GitLab. The page has a dark blue header with navigation links for 'Projects', 'Groups', and 'More'. A search bar is located on the right. The main content area is titled 'Create blank project' and contains a form with the following fields: 'Project name' (with the value 'laba-2'), 'Project URL' (with the value 'https://gitlab.com/' and a dropdown menu), 'Project slug' (with the value 'laba-2'), 'Project description (optional)' (with a text area containing 'Description format'), 'Visibility Level' (with radio buttons for 'Private' and 'Public', and 'Public' is selected), and 'Initialize repository with a README' (with a checkbox). Below the form are two buttons: 'Create project' and 'Cancel'.

3. Клонувати репозиторій собі на ПК двічі в різні директорії – це імітація роботи двох абсолютно різних розробників



```
git clone https://gitlab.com/d.kovalyuk/laba-2.git developer-1
```

```
git clone https://gitlab.com/d.kovalyuk/laba-2.git developer-2
```



Що потрібно зробити	Репозиторій	команда Git
Задати текстовий редактор (він буде потрібен для написання повідомлень про злиття)		git config
Додати файл: students.txt	developer-1	add, commit
Відправити «запустити» гілку master на сервер		git push origin
Перевірити через web-інтерфейс файл на сервері		
Забрати гілку master	developer-2	git pull
Створити нову гілку feature/64-login-page	developer-2	git branch
Переключитися в гілку feature/64-login-page	developer-2	git checkout
Додати та зберегти файл login.html	developer-2	add, commit
Відправити гілку feature/64-login-page на сервер	developer-2	git push origin
Почати відслідковувати feature/64-login-page	developer-1	git checkout feature/64-login-page
Додати та зберегти файл login.php	developer-1	add, commit
Відправити гілку feature/64-login-page на сервер	developer-1	git push origin feature/64-login-page
Почати відслідковувати feature/64-login-page	developer-2	git branch --set- upstream- to=origin/feature/64- login-page
Оновити гілку feature/64-login-page	developer-2	git pull
Влити гілку feature/64-login-page в master	developer-2	git merge
Відправити гілку master на сервер	developer-2	git push
Видалити гілку feature/64-login-page	developer-2	git branch -d
Видалити гілку feature/64-login-page на сервері	developer-2	\$git push origin -- delete feature/64- login-page

Додати файл: students.txt, відправити «запустити» гілку master на сервер

```
MINGW64:/d/git/developer-1

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git add students.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git commit -m 'add students list'
[master (root-commit) 27cd134] add students list
1 file changed, 6 insertions(+)
create mode 100644 students.txt

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
```

Безпека у Windows

Git Credential Manager

Enter credentials for 'https://gitlab.com/'

```
MINGW64:/d/git/developer-1

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git push origin
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 149.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/d.kovalyuk/laba-2.git
 * [new branch]      master -> master

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$
```

Перевірити через web-інтерфейс файл на сервері

The screenshot shows the GitLab web interface for a project named 'laba-2'. At the top, there is a notification bar stating 'Profile was successfully updated'. Below this, the project name 'laba-2' is displayed with its Project ID '24495789'. To the right of the project name are buttons for 'Star' (0) and 'Fork' (0). Below the project name, statistics are shown: '1 Commit', '1 Branch', '0 Tags', '133 KB Files', and '133 KB Storage'. A section for 'Auto DevOps' is visible, stating it will automatically build, test, and deploy the application based on a predefined CI/CD configuration. Below this, there are buttons for 'History', 'Find file', 'Web IDE', and 'Clone'. A commit titled 'add students list' by 'Dmytro Kovalyuk' is shown, authored 4 minutes ago. Below the commit, there are buttons for 'Add README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', and 'Set up CI/CD'. A table at the bottom shows the commit details:

Name	Last commit	Last update
students.txt	add students list	4 minutes ago

Забрати гілку master

```
MINGW64:/d/git/developer-2

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 278 bytes | 0 bytes/s, done.
From https://gitlab.com/d.kovalyuk/laba-2
* [new branch]      master    -> origin/master
```

Переключитися в гілку feature/64-login-page, додати та зберегти файл login.html, відправити гілку feature/64-login-page на сервер

```
MINGW64:/d/git/developer-2
$ git add login.html

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$ git commit -m 'add login form'
[feature/64-login-page 631c8f3] add login form
1 file changed, 1 insertion(+)
create mode 100644 login.html

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$ git push origin feature/64-login-page
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
```

```
MINGW64:/d/git/developer-1
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 287 bytes | 0 bytes/s, done.
From https://gitlab.com/d.kovalyuk/laba-2
* [new branch]      feature/64-login-page -> origin/feature/64-login-page
Already up to date.

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git branch
* master
```

Почати відслідковувати feature/64-login-page

```
MINGW64:/d/git/developer-1
$

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git branch
* master

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (master)
$ git checkout feature/64-login-page
Switched to a new branch 'feature/64-login-page'
Branch 'feature/64-login-page' set up to track remote branch 'feature/64-login-page' from 'origin'.
```

Додати та зберегти файл login.php, відправити гілку feature/64-login-page на сервер

```
MINGW64:/d/git/developer-1
$ git add login.php

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (feature/64-login-page)
$ git commit -m 'add login.php'
[feature/64-login-page fafd089] add login.php
1 file changed, 1 insertion(+)
create mode 100644 login.php

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-1 (feature/64-login-page)
$ git push origin feature/64-login-page
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
```

Почати відслідковувати feature/64-login-page

```
MINGW64:/d/git/developer-2
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$ git branch --set-upstream-to=origin/feature/64-login-page
Branch 'feature/64-login-page' set up to track remote branch 'feature/64-login-page' from 'origin'.

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$
```

Оновити гілку feature/64-login-page

```
MINGW64:/d/git/developer-2
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$ git pull
Updating 631c8f3..fafd089
Fast-forward
 login.php | 1 +
1 file changed, 1 insertion(+)
create mode 100644 login.php
```

Влити гілку feature/64-login-page в master, відправити гілку master на сервер

```
MINGW64:/d/git/developer-2
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (feature/64-login-page)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ git merge feature/64-login-page
Updating 27cd134..fafd089
Fast-forward
 login.html | 1 +
 login.php  | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 login.html
 create mode 100644 login.php

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/d.kovalyuk/laba-2.git
 27cd134..fafd089 master -> master

Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ |
```

Видалити гілку feature/64-login-page

```
MINGW64:/d/git/developer-2
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ git branch -d feature/64-login-page
Deleted branch feature/64-login-page (was fafd089).
```

Видалити гілку feature/64-login-page на сервері

```
MINGW64:/d/git/developer-2
Dima@DESKTOP-3BTJIEQ MINGW64 /d/git/developer-2 (master)
$ git push origin --delete feature/64-login-page
To https://gitlab.com/d.kovalyuk/laba-2.git
 - [deleted]          feature/64-login-page
```

Практична робота № 4

Патерни проектування

Мета роботи – засвоїти основні шаблони проектування, вміти їх використовувати для розв’язання типових задач.

Хід роботи

1. Визначити область застосування шаблону
2. Навести постановку задачі, яку розв’язує відповідний шаблон
3. Навести послідовність реалізації шаблону
4. Навести UML-діаграму взаємодії класів
5. Виконати програмну реалізацію шаблону.

Номер бригади	Шаблон
1	Будівельник (Builder)
2	Прототип (Prototype)
3	Фільтр (Filter)
4	Фасад (Facade)
5	Замісник (Proxy)
6	Команда (Comand)
7	Знімок (Memento)
8	Стратегія (Strategy)
9	Відвідувач (Visitors)

Практична робота № 5

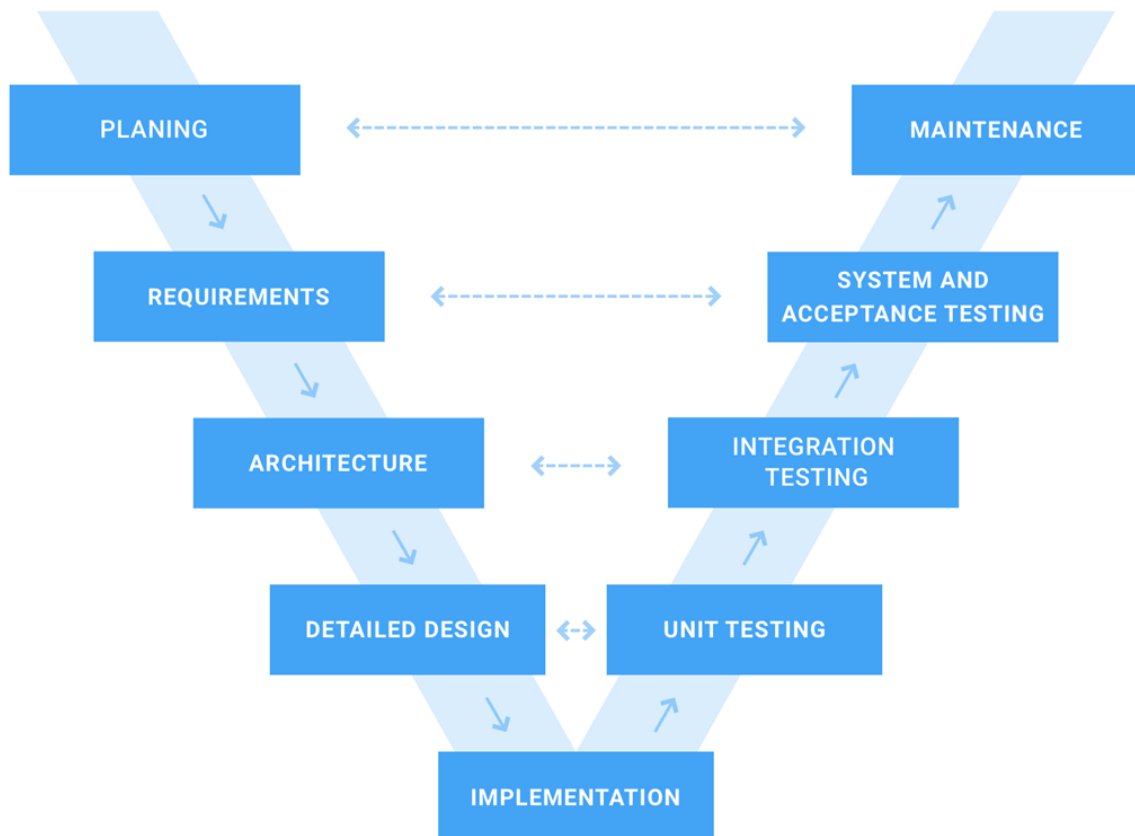
Модульне тестування

Мета роботи – опанувати методику проведення модульного тестування з використанням PHPUnit.

Теоретичні відомості

Тестування програмного забезпечення - це процес оцінки та перевірки певної системи програмного забезпечення, виявлення відмінностей між фактичним станом системи, бізнес вимогами та очікуваною поведінкою. Тестування вимірює загальну якість системи з точки зору: правильності, повноти, зручності використання, продуктивності для кінцевого користувача.

Одним із принципів організації тестування є «раннє тестування (early testing)», тобто проведення тестів якомога раніше, наскільки це можливо. V-модель розробки програмного забезпечення в повній мірі реалізує цей принцип



Якщо говорити про класифікацію тестів за рівнями, то виділяють наступні:

- модульне тестування (unit testing)
- інтеграційне тестування (integration testing)
- системне тестування (system testing)
- приймальне тестування (acceptance testing)

Модульне тестування - передбачає тестування окремих модулів і компонентів програми. Зазвичай юніт-тестуванням займаються самі розробники програмісти, використовуючи методи відлагодження коду, спеціалізовані бібліотеки та спеціальні фреймворки.

Інтеграційне тестування - спрямоване на перевірку взаємодії між модулями, компонентами програми, а також перевірку взаємодії програми з різними частинами системи (операційною системою, обладнанням, сервером, пристроями, іншими зв'язками у системі).

Системне тестування – основним завданням є перевірка як функціональних так і не функціональних вимог у системі в цілому.

Приймальне тестування - це фінальний етап тестування, яке проводиться спільно з кінцевим користувачем або клієнтом.

Хід роботи

Створити клас згідно варіанту.
Створити 2 тести для цього класу – один з яких виконається, інший ні.
Використати провайдер даних (Data Provider) – для генерації 4 наборів вхідних даних, один з яких – буде fail
Створити тест, який буде використовувати заглушку (mock)
Перевірити статус цих файлів
Виконати тести з консолі та IDE

Приклад виконання

1. Встановлюємо PHPUnit

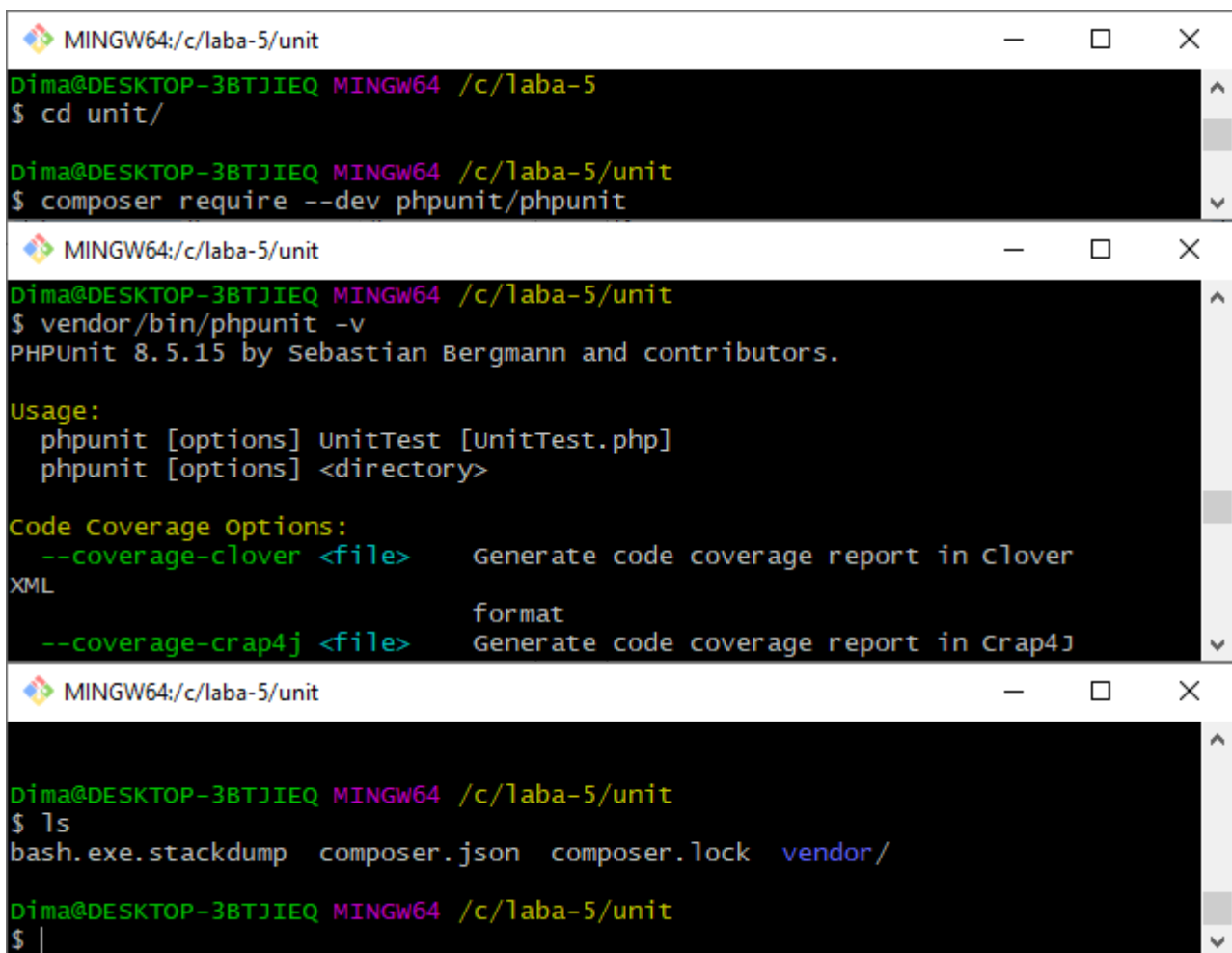
Офіційна документація: <https://phpunit.readthedocs.io/ru/latest/installation.html>

Встановлення з використанням composer

```
composer require --dev phpunit/phpunit
```

Перевірка версії PHPUnit – фактично перевірка правильності встановлення

```
vendor/bin/phpunit -v
```



The image shows three sequential screenshots of a terminal window with the title bar 'MINGW64:/c/laba-5/unit'. The first screenshot shows the command `cd unit/` being executed. The second screenshot shows the command `composer require --dev phpunit/phpunit` being executed, followed by the output of `vendor/bin/phpunit -v`, which displays the PHPUnit version (8.5.15) and usage instructions. The third screenshot shows the command `ls` being executed, listing the files in the current directory: `bash.exe.stackdump`, `composer.json`, `composer.lock`, and `vendor/`.

```
MINGW64:/c/laba-5/unit
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5
$ cd unit/

Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ composer require --dev phpunit/phpunit

MINGW64:/c/laba-5/unit
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ vendor/bin/phpunit -v
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

Usage:
 phpunit [options] UnitTest [UnitTest.php]
 phpunit [options] <directory>

Code Coverage Options:
 --coverage-clover <file>   Generate code coverage report in Clover
XML
                             format
 --coverage-crap4j <file>   Generate code coverage report in Crap4j

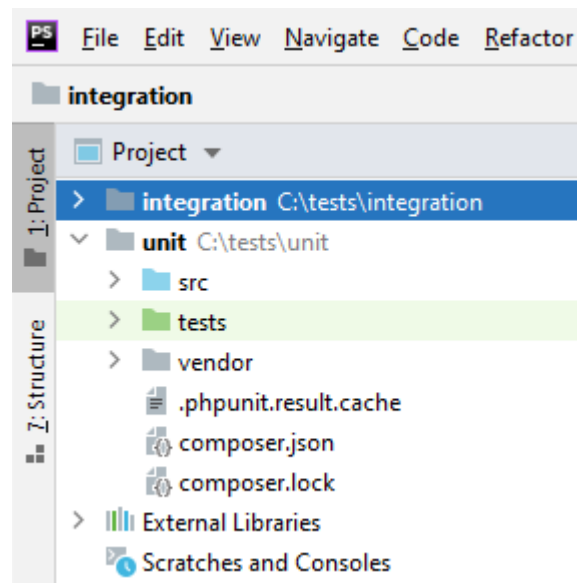
MINGW64:/c/laba-5/unit
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ ls
bash.exe.stackdump  composer.json  composer.lock  vendor/

Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ |
```

2. Створюємо структуру проекту

Рекомендована структура коду проекту і тестів

<https://phpunit.readthedocs.io/ru/latest/organizing-tests.html>



Наприклад

src	tests
~-- Currency.php	~-- CurrencyTest.php
~-- IntlFormatter.php	~-- IntlFormatterTest.php
~-- Money.php	~-- MoneyTest.php
~-- autoload.php	

Запускаємо тести

vendor/bin/phpunit tests/

Оскільки тестів ще немає, то отримаємо наступну відповідь

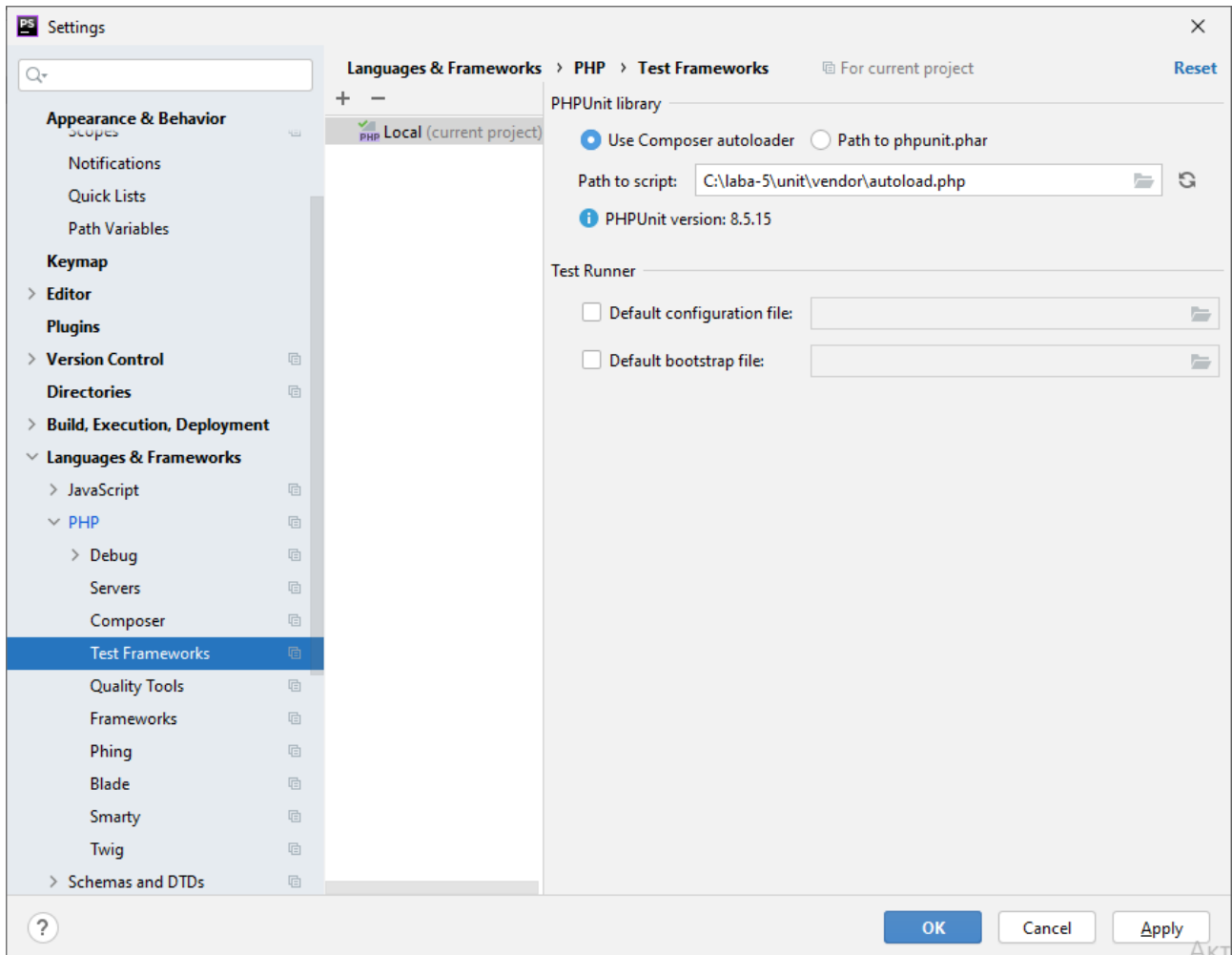
```
MINGW64:/c/laba-5/unit
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ vendor/bin/phpunit tests/
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.

No tests executed!

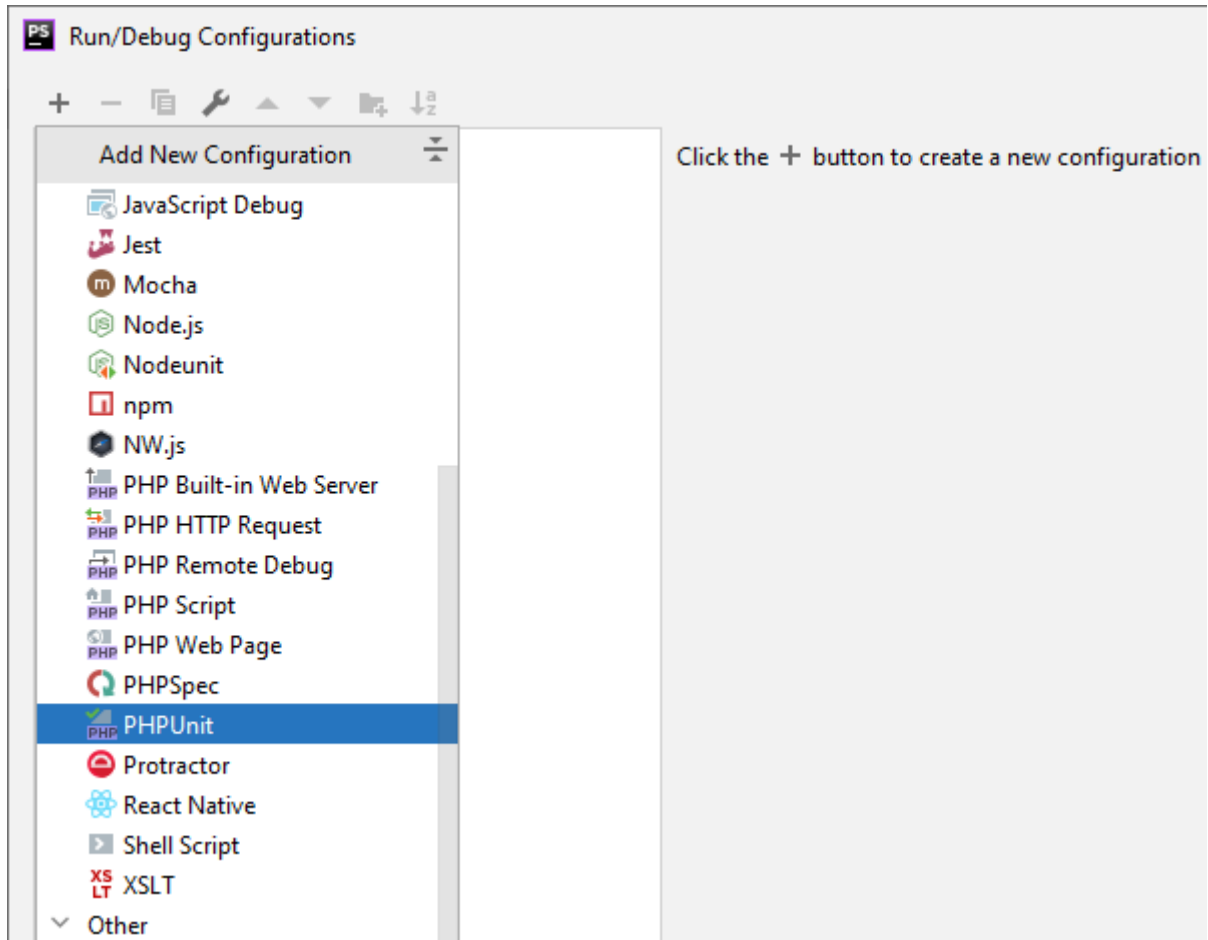
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ |
```

3. Налаштовуємо запуск тестів з IDE середовища

В якості IDE використовується PhpStorm. В налаштуваннях проекту необхідно вказати що використовується бібліотека PHPUnit і шлях до файлу autoload.php



Створюємо конфігурацію запуску тестів



Запускаємо (тестів поки немає)

```
Testing started at 12:11 ...  
C:\OpenServer\modules\php\PHP_7.4\php.exe C:/laba-5/unit/vendor/phpunit,  
PHPUnit 8.5.15 by Sebastian Bergmann and contributors.  
  
No tests executed!  
  
Process finished with exit code 0
```

Практична робота № 6

Інтеграційне тестування

Мета роботи – опанувати методику проведення інтеграційного тестування web-додатків з використанням Codeception.

Інтеграційне тестування - спрямоване на перевірку взаємодії між модулями, компонентами програми, а також перевірку взаємодії програми з різними частинами системи (операційною системою, обладнанням, сервером, пристроями, іншими зв'язками у системі).

Хід роботи

Створити php-скрипт(и), який відображає html сторінку, на основі методу Get з параметрами, отримуючи інформацію з БД.
Створити тест, який викликає цю сторінку з певними параметрами get-запиту і перевіряє, чи є на сторінці відповідна інформація (наприклад, Н1).
Створити тест, який робить запит в БД - операції вставки і вибору. Операції можна робити напрямку або через модель. Перевірити, чи з'явилася в БД інформація після операції Insert
Отримати звіт про виконання тестів у вигляді html сторінки

Приклад виконання

Хід роботи

1. Встановлюємо Codeception (<https://codeception.com>)
<https://codeception.com/quickstart>

```
composer require "codeception/codeception" --dev  
vendor/bin/codecept bootstrap
```

```
MINGW64:/c/laba-5/integration
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/unit
$ cd ../integration/

Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/integration
$ composer require "codeception/codeception" --dev
Warning from https://repo.packagist.org: You are using an outdated version of
Composer. Composer 2 is now available and you should upgrade. See https://getcomposer.org/2
Using version ^4.1 for codeception/codeception
```

```
MINGW64:/c/laba-5/integration
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/integration
$ ls
composer.json  composer.lock  vendor/

Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/integration
$
```

```
MINGW64:/c/laba-5/integration
Dima@DESKTOP-3BTJIEQ MINGW64 /c/laba-5/integration
$ vendor/bin/codecept bootstrap
Bootstrapping Codeception

File codeception.yml created      <- global configuration
Adding codeception/module-phpbrowser for PhpBrowser to composer.json
Adding codeception/module-asserts for Asserts to composer.json
2 new packages added to require-dev
? composer.json updated. Do you want to run "composer update"? (y/n) y
Running composer update
Loading composer repositories with package information
```

Запускаємо тести (тестів ще немає)

```
vendor/bin/codecept run --steps
```

Створюємо клас для інтеграційних тестів (в термінології Codeceptor вони розглядаються як функціональні)

```
vendor/bin/codecept generate:cept functional students
```

В класі створюємо два тести

1. Перевірка взаємодії з базою даних (використаємо таблицю groups)

	group_id	state	kurs	title	type
1	1	денна	5	ЛА-61	1
2	2	денна	2	ЛА-62	1
3	3	заочна	1	ЛА-301	2

2. Перевірка відображення веб-сторінки для запиту з параметрами

← → ↻ 🏠 ⚠ Не захищено | lections.tprz/decanat/students.php?group_title=la61&sort_id=1

група : сортувати :

Список групи : ЛА-61

Список студентів

Біганський Богдан Миколайович	98.7
Бондаренко Катерина Олександрівна	93
Бугаєнко Ігор Іванович	84.2
Варнов Олег Олександрович	84.2
Герасименко Олександр Сергійович	80

Встановлюємо модуль для роботи з БД

```
composer require --dev codeception/module-db
```

Задаємо параметри підключення до БД і домен сайту у файлі налаштувань functional.suite.yml

```
actor: FunctionalTester
modules:
  enabled:
    # add a framework module here
    - \Helper\Functional

    - PhpBrowser:
        url: http://lections.tprz

    - Db:
        dsn: 'mysql:host=localhost;dbname=kpi'
        user: 'root'
        password: 'root'
        cleanup: false
```


Клас тестів має наступний вигляд

```
class StudentsCest
{
    public function _before(FunctionalTester $I)
    {
    }

    // tests
    public function testStudentsInGroup(FunctionalTester $I)
    {
        $I->amOnPage('/decanat/students.php?group_title=la61&sort_id=1');
        $I->canSee('Список групи : ЛА-61', 'p');
    }

    // tests
    public function testgetGroupByTitle(FunctionalTester $I)
    {
        $I->haveInDatabase('groups',
            array('title' => 'ЛА-92',
                'kurs' => '2',
                'type' => '1',
                'state' => 'МОЛОДШИЙ'));

        $I->seeInDatabase('groups', ['title' => 'ЛА-92']);
    }
}
```

Запускаємо тести

```
vendor/bin/codecept run --html
```

Практична робота № 7

GitLab CI/CD

Мета роботи – опанувати методику неперервної інтеграції та розгортання програмного забезпечення в середовищі GitLab на прикладі статичного web-сайту.

Теоретичні відомості

- https://docs.gitlab.com/ee/ci/quick_start/
- https://docs.gitlab.com/ee/user/project/pages/getting_started/pages_from_scratch.html
- <https://docs.gitlab.com/ee/user/project/pages/>

Хід роботи

1. Опанувати директиви конфігураційного файлу, ознайомитися з поняттям стадій (stages) та дій (jobs), які виконуються на певних стадіях.

Створити файл конфігурації `.gitlab-ci.yml`, що визначає послідовність дій в pipeline та містить наступний код:

```
build-job:
  stage: build
  script:
    - echo "Hello, $GITLAB_USER_LOGIN!"

test-job1:
  stage: test
  script:
    - echo "This job tests something"

test-job2:
  stage: test
  script:
    - echo "This job tests something, but takes more time than test-job1."
    - echo "After the echo commands complete, it runs the sleep command for 20 seconds"
    - echo "which simulates a test that runs 20 seconds longer than test-job1"
    - sleep 20

deploy-prod:
  stage: deploy
  script:
    - echo "This job deploys something from the $CI_COMMIT_BRANCH branch."
```

Зберегти файл, подивитися вкладку Pipeline, Jobs.

gitlab.com/-/ide/project/d.kovalyuk/laba-7-1/edit/master/-/

GitLab Projects Groups More Search or jump to...

laba-7-1
d.kovalyuk/laba-7-1

Edit

File templates .gitlab-ci.yml Choose a templat...

```

1 build-job:
2   stage: build
3   script:
4     - echo "Hello, $GITLAB_USER_LOGIN!"
5
6 test-job1:
7   stage: test
8   script:
9     - echo "This job tests something"
10
11 test-job2:
12   stage: test
13   script:
14     - echo "This job tests something, but takes more time than test-job1."
15     - echo "After the echo commands complete, it runs the sleep command for 20 seconds"
16     - echo "which simulates a test that runs 20 seconds longer than test-job1"
17     - sleep 20
18
19 deploy-prod:
20   stage: deploy
21   script:
22     - echo "This job deploys something from the $CI_COMMIT_BRANCH branch."
23

```

gitlab.com/d.kovalyuk/laba-7-1/-/pipelines

GitLab Projects Groups More Search or jump to...

laba-7-1

Project overview
Repository
Issues 0
Merge requests 0
Requirements
CI/CD
Pipelines
Editor

Dmytro Kovalyuk > laba-7-1 > Pipelines

All 1 Finished Branches Tags

Filter pipelines

Status	Pipeline	Triggerer	Commit	Stages	Duration
running	#294219103 latest		master → b43e66f4 Update .gitlab-ci.yml	test: running	In progress

2. Створити pipeline для CI та CD статичного сайту.

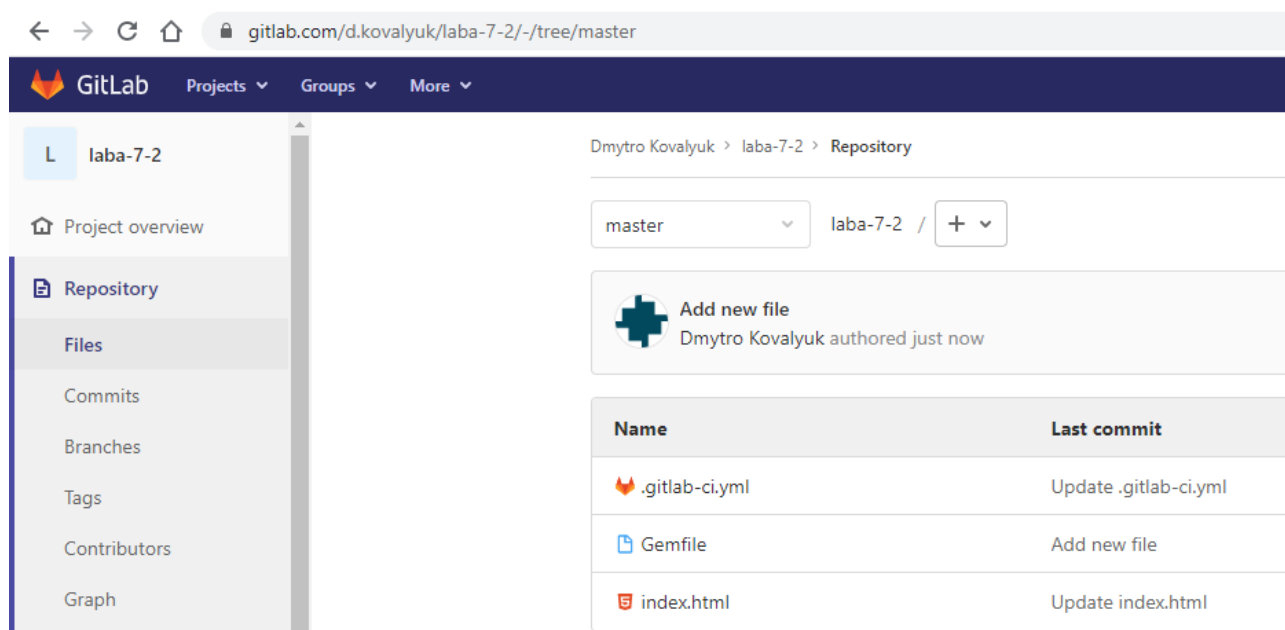
Основні вимоги:

- Для створення сайту використовується статичний генератор Jekyll (<https://jekyllrb.com/>)
- Для хостингу використовується вбудований репозиторій GitLab Pages (<https://docs.gitlab.com/ee/user/project/pages/>)
- Тестування (unit) здійснюється для гілок розробки (not master), публікація нового контенту на GitLab Pages здійснюється тільки при оновленні гілки master.

Створити в репозиторії наступні файли

- `.gitlab-ci.yml` - Файл YAML, що містить команди, які потрібно виконати.
- `index.html` - HTML-файл, з будь-яким вмістом
- `Gemfile` - Файл, що описує залежності для програм Ruby. Заповніть його наступним вмістом:

```
source "https://rubygems.org"
gem "jekyll"
```



Додати наступний код в `.gitlab-ci.yml`

```
image: ruby:2.7
workflow:
  rules:
    - if: '$CI_COMMIT_BRANCH'

cache:
  paths:
    - vendor/

before_script:
  - gem install bundler
  - bundle install --path vendor

pages:
  stage: deploy
  script:
    - bundle exec jekyll build -d public
  artifacts:
    paths:
      - public
  rules:
    - if: '$CI_COMMIT_BRANCH == "master"'

test:
  stage: test
  script:
    - bundle exec jekyll build -d test
  artifacts:
    paths:
      - test
  rules:
    - if: '$CI_COMMIT_BRANCH != "master"'
```

Запустити ріплайн, подивитися результат розгортання сайту

The screenshot displays the GitLab CI/CD interface. On the left, a sidebar contains navigation links: Requirements, CI/CD, Security & Compliance, Operations, Packages & Registries, Analytics, Wiki, Snippets, Members, and Settings. The 'Settings' link is selected. A sub-menu is open, showing options: General, Integrations, Webhooks, Repository, CI/CD, Operations, and Pages. The 'CI/CD' option is selected. The main content area shows the pipeline configuration for the 'master' branch on the 'laba-8-2' runner. A commit by 'Dmytro Kovalyuk' is shown, with a 'change name' action. Below the commit, there are buttons for 'Upload File', 'CI/CD configuration', 'Add README', 'Add LICENSE', 'Add CHANGELOG', 'Add Kubernetes cluster', and 'Configure Integrations'. A table lists the files in the pipeline:

Name	Last commit
.gitlab-ci.yml	Update .gitlab-ci.yml
Gemfile	Add new file
index.html	change name

Робота з створеним CI/CD.

- Клонувати репозиторій з GitLab на локальний ПК.
- Локально створити гілку feature/name – внести зміни в html-файл.
- Відправити гілку feature/name в репозиторій GitLab, подивитися результат роботи pipeline, подивитися вміст сайту за url-адресою на Pages.
- Локально влити гілку feature/name в master. Відправити гілку master в репозиторій GitLab, подивитися результат роботи pipeline, подивитися вміст сайту за url-адресою на Pages.

Літературні джерела

1. Jira tutorials: Learn agile with Jira Software. Режим доступу - <https://www.atlassian.com/agile/tutorials>
2. Pro Git book, written by Scott Chacon and Ben Straub - Друге видання. Режим доступу українською мовою: <https://git-scm.com/book/uk/v2>
3. Олександр Швець Рефакторинг, патерни проектування, принципи SOLID – Режим доступу: <https://refactoring.guru/uk>
4. Design Patterns in Java Tutorial. – Режим доступу: tutorialspoint.com/design_pattern
5. Керівництво по PHPUnit – Режим доступу: <https://phpunit.readthedocs.io/ru/latest/index.html>
6. Офіційна документація Codeception – Режим доступу: <https://codeception.com/quickstart>
7. Офіційна документація GitLab CI/CD – Режим доступу: <https://docs.gitlab.com/ee/ci/>
8. The Java™ Tutorials – Режим доступу: <https://docs.oracle.com/javase/tutorial/>
9. Офіційна документація PHP – Режим доступу: <http://php.net/manual/ru>
10. Офіційна документація Python – Режим доступу: <https://docs.python.org/3/>