

## **ДОДАТОК А**

Графічний матеріал кваліфікаційної роботи

Харківський національний університет  
радіоелектроніки

Кафедра КІТС

Кваліфікаційна робота на тему:  
**Інтелектуальний мультизадачний персональний  
AI-асистент на основі Model Context Protocol**

Виконав: студент гр. КІУКІ-21-10  
Попов Артем

Керівник доц. Сердюк Н.М.

## Мета кваліфікаційної роботи

- Проектування та розробка мультизадачного персонального AI-асистенту, що матиме можливість інтеграції з зовнішньої системою у реальному часі.



## Вступ

- ▶ Класичні асистенти зазвичай обмежені в плані взаємодії з оточенням, проте постійно з'являються нові інструменти, спрямовані на полегшення інтеграції асистентів в повсякденне життя.
- ▶ Завдяки подібним технологіям стає можливою автоматична або контрольована взаємодія асистентів з зовнішніми системами.



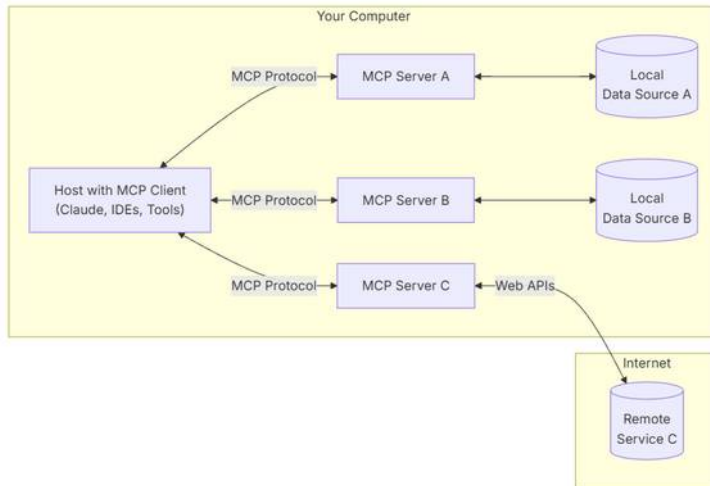
## Існуючі способи інтеграції ШІ в існуючі рішення

- ▶ REST API
- ▶ Retrieval Augmented Generation
- ▶ Модульні фреймворки
- ▶ Model Context Protocol



LangChain

## Model Context Protocol



5

## JSON RPC

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23, "minuend": 42}, "id": 3}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 3}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"minuend": 42, "subtrahend": 23}, "id": 4}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 4}
```

6

## Портал DL NURE

DL NURE UA

На головну Особистий кабінет **Міі курси**



### Міі курси

Огляд курсу

В процесі - Знайти Сопутувати за останнім доступом - Список -

|  |   |  |
|--|---|--|
|  | І КТРС БАКХКХУ/ІПРО а24/25<br>Паралельні та розподілені обчислення<br>2024-2025<br>0% завершено |  |
|  | І КТРС БАКХКХУ/ІВДІ а24/25<br>*Вступ до DevOps<br>2024-2025<br>0% завершено                     |  |
|  | І КТРС БАКХКХУ/ІМА а24/25<br>*Мікросервісній архітектурі<br>2024-2025<br>0% завершено           |  |
|  | І ЕОМ БАКХКХУ/ІХМ а24/25<br>Комп'ютерна мережі<br>2024-2025<br>0% завершено                     |  |
|  | І КТРС БАКХКХУ/ІОСВ а24/25<br>*Основи Computer Vision<br>2024-2025<br>0% завершено              |  |

7

## Можливості навчальної платформи DL NURE

- ▶ перегляд переліку поточних курсів, на які записаний студент
- ▶ перегляд матеріалів конкретних курсів (посилання на лекції або лабораторні роботи, тести, методичні вказівки)
- ▶ завантаження самостійно виконаних робіт для подальшої перевірки викладачами та отримання оцінки
- ▶ проходження тестів з різних дисциплін
- ▶ самостійне ведення відвідування для деяких дисциплін

8

## Можливості персонального асистенту

- ▶ відкривати портал DL NURE
- ▶ відповідати на запитання про поточні курси користувача
- ▶ відповідати на запитання про вміст окремих курсів
- ▶ повертати посилання на відповідні ресурси при запиті користувача
- ▶ надавати стислий вміст документів, наявних на порталі

## Технології реалізації MCP сервера

- ▶ Мова програмування Java
- ▶ Фреймворк Spring AI
- ▶ Бібліотека Microsoft Playwright
- ▶ OpenAI API



## Інтерфейс наданий створеним MCR сервером

- ▶ `openDL` - запуск порталу DL NURE
- ▶ `getEnrolledCourses` - отримати перелік доступних курсів
- ▶ `getCourseItems` - отримати вміст курсу
- ▶ `getPdfSummary` - отримати стислий вміст PDF-документу

## Структуризація даних

\*Основи Computer Vision

**Відвідування**

Відвідування  
Відвідуваність за семестр: 2  
Бали по всіх відвідуваннях за семестр: 4 / 4  
Відсоток по всіх відвідуваннях за семестр: 100.0%

Загальна кількість занять: 15  
Бали по всіх заняттях: 4 / 30  
Відсоток по всіх заняттях: 13.3%

Максимально можливі бали: 30 / 30  
Максимально можлива відсоток: 100.0%

**Загальне** Згорнути все

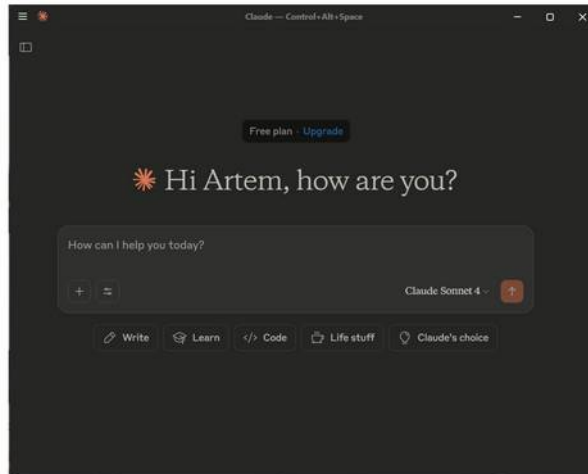
- Оголошення
- Відвідування
- Новини

**Консультаційний пункт**

- Лещі онлайн (Google Hangouts Meet)  
За розкладом консультацій на заняття надаватиметься до сайту відеоконференції
- Лабораторні роботи  
За графіком ІІ-семестру можна знайти розклад

| Назва                                | Тип ресурсу          | Опис                         | Посилання   |
|--------------------------------------|----------------------|------------------------------|---|
| Оголошення                           | Announcement / Forum |                              | <a href="https://dl.nure.ua/mod/forum/view.php?id=647664">https://dl.nure.ua/mod/forum/view.php?id=647664</a> |
| Відвідування                         | Attendance           |                              | Посилання на ресурс   |
| Новини                               | Announcement / Forum |                              | Посилання на ресурс   |
| Лекції online (Google Hangouts Meet) | Link                 | За розкладом консультацій... | Посилання на ресурс   |
| Лабораторні роботи                   | Link                 | За графіком ЛБ...            | Посилання на ресурс   |

## Claude Desktop в якості MCP клієнта



13

## Інтеграція створеного MCP серверу у Claude Desktop

```
{
  "mcpServers": {
    "dl-mcp": {
      "command": "java",
      "args": [
        "-Ddl.login=userlogin",
        "-Ddl.password=userpassword",
        "-DOPENAI_API_KEY=apikey",
        "-jar",
        "D:/Projects/dl-assistant/target/dl-assistant-0.0.1-SNAPSHOT.jar"
      ]
    }
  }
}
```

14

## Демонстрація

**A** Розкажи будь ласка про що 3 лабораторна робота з ПТРО?

Для того, щоб знайти інформацію про 3 лабораторну роботу з ПТРО (Паралельні та розподілені обчислення), давайте спочатку перевіримо детальну інформацію про цей курс:

```
G getCourseItems
```

Request

```
{
  "courseTitle": "Паралельні та розподілені обчислення"
}
```

Response

```
[{"title": "Відвідування ЗАЛІК", "type": "Attendance", "description": "", "url": "https://dl.nure.ua/mod/attendance/view?id=672617"}, {"title": "Правила використання ШІ під час навчання", "type": "PDF"}]
```

Чудово! Я знайшов методичні вказівки для лабораторної роботи з ПТРО. Давайте отримаємо детальну інформацію про неї:

Claude — Control-Alt-Space

Recording has started

Free plan · Upgrade

What's new, Artem?

How can I help you today?

Claude 3.7 Sonnet

Write · Learn · Code · Life stuff · Claude's choice

11:13 11.01.2025

## Висновки

- ▶ Завдяки інтелектуальним асистентам можна оптимізувати повсякденні процеси що вимагають уваги та накладних витрат часу
- ▶ Було спроектовано мультизадачного персонального AI-асистента, що має інтеграцію з навчальною платформою у реальному часі.
- ▶ Створений асистент має спростити процес навігації навчальним порталом та пошуку інформації, дозволяючи зосередити увагу на більш важливих задачах.

17

Дякую за увагу!

18

**ДОДАТОК Б**

Код програми



```

        <artifactId>lombok</artifactId>
        <optional>>true</optional>
        </dependency>
    </dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    </dependency>
    </dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.ai</groupId>
            <artifactId>spring-ai-bom</artifactId>
            <version>${spring-ai.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <annotationProcessorPaths>
                        <path>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </path>
                    </annotationProcessorPaths>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

```

```
</project>
```

## Б.2 Клас DLAssistantApplication.java

```
package com.lezdoit.dlassistant;

import com.lezdoit.dlassistant.dl.DLService;
import org.springframework.ai.tool.ToolCallbackProvider;
import org.springframework.ai.tool.method.MethodToolCallbackProvider;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class DLAssistantApplication {

    public static void main(String[] args) {
        SpringApplication.run(DLAssistantApplication.class, args);
    }

    @Bean
    public ToolCallbackProvider browserTools(DLService dlService) {
        MethodToolCallbackProvider.builder().toolObjects(dlService).build();
    }
}
```

return

### Б.3 Клас DLService.java

```
package com.lezdoit.dlassistant.dl;

import com.lezdoit.dlassistant.tools.ai.AITool;
import com.lezdoit.dlassistant.tools.pdf.PDFProcessor;
import com.microsoft.playwright.Browser;
import com.microsoft.playwright.BrowserType;
import com.microsoft.playwright.Locator;
import com.microsoft.playwright.Page;
import com.microsoft.playwright.Playwright;
import com.microsoft.playwright.options.Cookie;
import jakarta.annotation.PostConstruct;
import jakarta.annotation.PreDestroy;
import lombok.RequiredArgsConstructor;
import org.springframework.ai.tool.annotation.Tool;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Lazy;
import org.springframework.core.io.Resource;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

@Service
@RequiredArgsConstructor
public class DLService {

    private final DLCredentials credentials;

    private AITool aiTool;

    @Autowired
    public void setAiTool(@Lazy AITool aiTool) {
        this.aiTool = aiTool;
    }

    private Playwright playwright;

    private Browser browser;

    private Page page;

    private final CourseItemTypeMatcher courseItemTypeMatcher;

    private final RestTemplate restTemplate;

    private final PDFProcessor pdfProcessor;

    @PostConstruct
    public void init() throws IOException, InterruptedException {
        playwright = Playwright.create();
        browser = playwright.chromium().launch(
            new BrowserType.LaunchOptions()
                .setHeadless(false)
        );
    }
}
```

```

        .setSlowMo(2000)
    );

    openDL();
    List<CourseItem> courseItems = getCourseItems("*Основи Computer Vision");

    System.out.println("mew");
}

@PreDestroy
public void destroy() {
    browser.close();
    playwright.close();
}

@Tool(description = ""
        Launches a browser and opens DL NURE portal.
        This action is required before conducting any other actions with DL
portal.
        "")
public String openDL() {
    page = browser.newPage();
    page.navigate(DLPages.LOGIN);

    page.locator("#username").waitFor();

    //enter the login and password
    page.locator("#username").fill(credentials.getLogin());
    page.locator("#password").fill(credentials.getPassword());

    //click the Log In button
    page.click("#loginbtn");

    return "Browser launched successfully and navigated to DL";
}

@Tool(description = ""
        Returns a list of the courses the user is enrolled to.
        "")
public List<String> getEnrolledCourses() {
    page.navigate(DLPages.COURSES);

    Locator courseTitles = page.locator(".aalink.coursename");

    List<String> titles = new ArrayList<>();

    for (Locator courseLocator : courseTitles.all()) {
        String fullText = courseLocator.innerText();
        String title = fullText.split("\n")[1].trim();
        titles.add(title);
    }

    return titles;
}

@Tool(description = ""
        Returns a list of all items present on a specific course page.
        "")
public List<CourseItem> getCourseItems(String courseTitle) throws
IOException, InterruptedException {
    openCoursePage(courseTitle);

    Locator courseItemsLocator = page.locator(".activity-item.focus-

```

```

control");

    List<CourseItem> courseItems = new ArrayList<>();

    for (Locator courseItemLocator : courseItemsLocator.all()) {
        CourseItem courseItem = new CourseItem();

        Locator titleLocator =
courseItemLocator.locator("span.instancename");
        if (!titleLocator.all().isEmpty()) {
            String title = (String) titleLocator
                .evaluate("node => node.childNodes[0] ?
node.childNodes[0].textContent.trim() : '');
            courseItem.setTitle(title);

            Locator typeLocator = courseItemLocator.locator(".activityicon");
            if (!typeLocator.all().isEmpty()) {
                String typeImgSrc = (String) typeLocator
                    .evaluate("node => node.getAttribute('src')");
                String type =
courseItemTypeMatcher.getTypeFromIconUrl(typeImgSrc);
                courseItem.setType(type);
            }

            Locator descriptionLocator =
courseItemLocator.locator(".activity-description, .activity-dates");
            if (!descriptionLocator.all().isEmpty()) {
                // Handle multiple elements by concatenating their text
                StringBuilder descriptionBuilder = new StringBuilder();
                for (Locator element : descriptionLocator.all()) {
                    if (descriptionBuilder.length() > 0) {
                        descriptionBuilder.append(" ");
                    }
                    descriptionBuilder.append(element.innerText());
                }
                courseItem.setDescription(descriptionBuilder.toString());
            }

            Locator urlLocator =
courseItemLocator.locator(".aalink.stretched-link");
            if (!urlLocator.all().isEmpty()) {
                String url = (String) urlLocator
                    .evaluate("node => node.getAttribute('href')");
                courseItem.setUrl(url);
            }

            courseItems.add(courseItem);
        }
    }

    return courseItems;
}

public void openCoursePage(String courseTitle) {
    page.navigate(DLPages.COURSES);

    Locator courseTitles = page.locator(".aalink.coursename");

    boolean courseFound = false;
    for (Locator courseLocator : courseTitles.all()) {
        String fullText = courseLocator.innerText();
        String title = fullText.split("\n")[1].trim();
    }
}

```

```

        if (title.equalsIgnoreCase(courseTitle)) {
            courseLocator.click();
            courseFound = true;
            break;
        }
    }

    if (!courseFound) {
        throw new IllegalArgumentException("Course not found: " +
courseTitle);
    }
}

@Tool(description = ""
        Downloads a PDF file from specified link, generates summary and
returns it as a plain text.
        "")
public String getPdfSummary(String pdfLink) throws IOException {
    // Get cookies from Playwright browser context
    List<Cookie> cookies = browser.contexts().get(0).cookies();

    // Create HttpHeaders and add cookies
    HttpHeaders headers = new HttpHeaders();

    // Convert Playwright cookies to a cookie header string
    StringBuilder cookieHeader = new StringBuilder();
    for (Cookie cookie : cookies) {
        if (!cookieHeader.isEmpty()) {
            cookieHeader.append("; ");
        }
        cookieHeader.append(cookie.name).append("=").append(cookie.value);
    }

    // Add the cookie header to the request
    headers.add("Cookie", cookieHeader.toString());

    // Create HttpEntity with headers
    HttpEntity<String> entity = new HttpEntity<>(headers);

    // Make the request with the entity containing cookies
    ResponseEntity<Resource> response = restTemplate.exchange(
        pdfLink,
        HttpMethod.GET,
        entity,
        Resource.class
    );

    // Process the PDF as before
    String plainText =
pdfProcessor.getPdfAsPlainText(response.getBody().getInputStream());

    return aiTool.getDocumentSummary(plainText);
}
}

```

## Б.4 Класи DLPages.java, DLCredentials.java, CourseItemTypeMatcher.java

```
package com.lezdoit.dlassistant.dl;

public class DLPages {
    public static final String LOGIN = "https://dl.nure.ua/login/index.php";
    public static final String COURSES = "https://dl.nure.ua/my/courses.php";
}
```

```
package com.lezdoit.dlassistant.dl;

import lombok.Getter;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Configuration;

@Configuration
@Getter
public class DLCredentials {

    @Value("${dl.login}")
    private String login;

    @Value("${dl.password}")
    private String password;

}
```

```
package com.lezdoit.dlassistant.dl;

import org.springframework.stereotype.Service;

@Service
public class CourseItemTypeMatcher {
    public String getTypeFromIconUrl(String url) {
        if (url.contains("forum")) {
            return "Announcement / Forum";
        } else if (url.contains("attendance")) {
            return "Attendance";
        } else if (url.contains("url")) {
            return "Link";
        } else if (url.contains("folder")) {
            return "Folder";
        } else if (url.contains("pdf")) {
            return "PDF Link";
        } else if (url.contains("quiz")) {
            return "Test";
        } else if (url.contains("document")) {
            return "Document";
        } else if (url.contains("assign")) {
            return "Assignment Upload";
        } else {
            return "Unknown";
        }
    }
}
```

## Б.7 Класи CourseItem.java, AITool.java

```
package com.lezdoit.dlassistant.dl;
```

```
import lombok.Data;
```

```
@Data
```

```
public class CourseItem {  
    private String title;  
    private String type;  
    private String description = "";  
    private String url = "";  
}
```

```
package com.lezdoit.dlassistant.tools.ai;
```

```
import lombok.RequiredArgsConstructor;  
import org.springframework.ai.openai.OpenAiChatModel;  
import org.springframework.stereotype.Service;
```

```
@Service
```

```
@RequiredArgsConstructor
```

```
public class AITool {
```

```
    private final OpenAiChatModel chatModel;
```

```
    public String getDocumentSummary(String documentContent) {
```

```
        // Create a prompt for document summarization
```

```
        String promptText = ""
```

```
            You are a helpful assistant that summarizes documents.
```

```
            Provide a concise summary of the document content provided by the
```

```
user. " +
```

```
            The summary should contain structure of the document. " +
```

```
            The summary should contain key points from the document content.
```

```
" +
```

```
            SUMMARY SHOULD BE PROVIDED IN ORIGINAL LANGUAGE OF THE DOCUMENT.
```

```
" +
```

```
            Please summarize the following document:
```

```
            "" + documentContent;
```

```
        return chatModel.call(promptText);
```

```
    }
```

```
}
```

## Б.9 Клас PDFProcessor.java

```
package com.lezdoit.dlassistant.tools.pdf;

import org.apache.pdfbox.Loader;
import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.text.PDFTextStripper;
import org.springframework.stereotype.Service;

import java.io.IOException;
import java.io.InputStream;
import java.util.regex.Pattern;

@Service
public class PDFProcessor {

    public String getPdfAsPlainText(InputStream inputStream) throws IOException {
        byte[] pdfData = inputStream.readAllBytes();

        if (!isPdfDataValid(pdfData)) {
            return "Error: The file does not appear to be a valid PDF document.";
        }

        try {
            try (PDDocument document = Loader.loadPDF(pdfData)) {
                String rawText = extractTextFromPdf(document);

                return optimizeText(rawText);
            } catch (IOException e) {
                try {
                    try (PDDocument document = Loader.loadPDF(pdfData, "")) {
                        String rawText = extractTextFromPdf(document);
                        return optimizeText(rawText);
                    }
                } catch (IOException fallbackException) {
                    try {
                        System.setProperty("org.apache.pdfbox.baseParser.pushBackSize", "1000000");
                        try (PDDocument document = Loader.loadPDF(pdfData)) {
                            String rawText = extractTextFromPdf(document);
                            return optimizeText(rawText);
                        }
                    } catch (IOException lastAttemptException) {
                        return "Error: Unable to process PDF. The file may be corrupted or password-protected. " +
                            "Original error: " + e.getMessage();
                    } finally {
                        System.clearProperty("org.apache.pdfbox.baseParser.pushBackSize");
                    }
                }
            }
        }

        private boolean isPdfDataValid(byte[] data) {
            if (data == null || data.length < 5) {
                return false;
            }
        }
    }
}
```

```

String signature = new String(data, 0, Math.min(5, data.length));
    return signature.startsWith("%PDF-");
}

private String extractTextFromPdf(PDDocument document) throws IOException {
    PDFTextStripper stripper = new PDFTextStripper();
    // Set reading order to left to right
    stripper.setSortByPosition(true);
    // Extract text from all pages
    return stripper.getText(document);
}

private String optimizeText(String text) {
    if (text == null || text.isEmpty()) {
        return "";
    }

    String optimized = text.replaceAll("\\s+", " ");

    optimized = optimized.replaceAll("\\n\\s*\\n", "\\n\\n");
    optimized = optimized.replaceAll("\\n(?:!\\n)", " ");
    optimized = optimized.replaceAll("(?i)\\bpage\\s+\\d+\\s+of\\s+\\d+\\b",
    "");
    optimized = optimized.replaceAll("(?m)^\\s*\\d+\\s*$", "");
    optimized = removeRepeatedHeadersFooters(optimized);
    optimized = optimized.trim();

    return optimized;
}

private String removeRepeatedHeadersFooters(String text) {
    String[] lines = text.split("\\n");
    if (lines.length <= 1) {
        return text;
    }

    Pattern shortLinePattern = Pattern.compile("^.{1,60}$");

    StringBuilder result = new StringBuilder();
    for (String line : lines) {
        String trimmedLine = line.trim();
        if (trimmedLine.isEmpty()) {
            result.append("\\n");
            continue;
        }

        if (shortLinePattern.matcher(trimmedLine).matches() &&
            countOccurrences(text, trimmedLine) > 2) {
            continue;
        }
        result.append(line).append("\\n");
    }
    return result.toString();
}

private int countOccurrences(String text, String searchString) {
    int count = 0;
    int index = 0;
    while ((index = text.indexOf(searchString, index)) != -1) {
        count++;
        index += searchString.length();
    }
    return count;
}

```

} }