

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ННК «ІПСА» НТУУ «КПІ ІМ. ІГОРЯ СІКОРСЬКОГО»**

**КАФЕДРА ММСА**

**Проект**

**З дисципліни: Комп’ютерні мережі**

***Власний HTTP сервер***

**Виконала:**

**Студентка ІII курсу**

**Групи КА-77**

**Пастушок О. О.**

**Перевірив: Кухарєв С. О.**

**Київ 2020**

**Мета проекту:** За допомогою власного програмного продукту продемонструвати роботу з протоколами, які передаються на технічному рівня програми під час надсилання запитів та отримання на них відповідей.

**План реалізації проекту:**

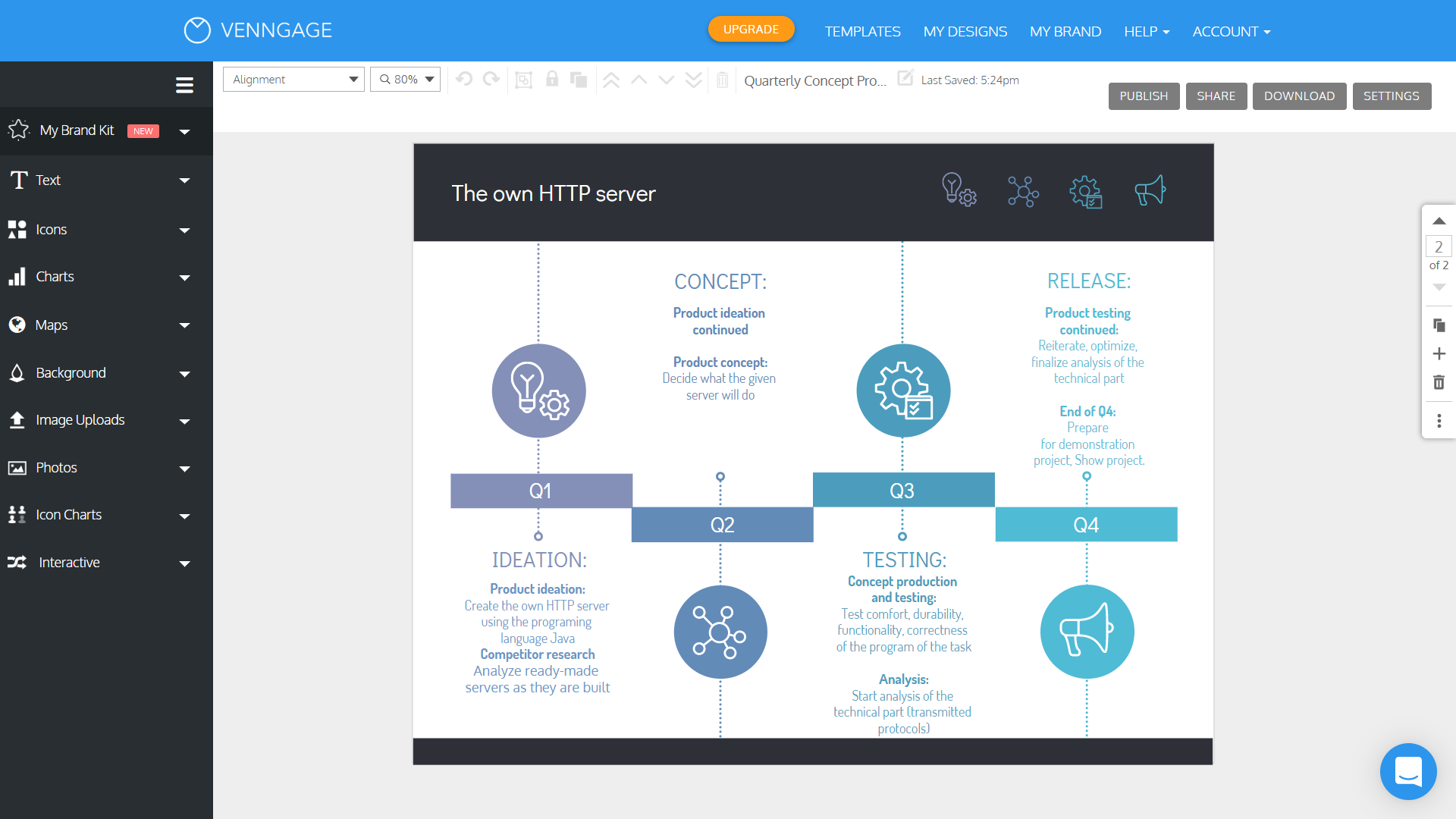
1. За допомогою мови програмування Java написати HTTP сервер, який прийматиме запити та буде давати відповідь на даний запит.
2. Дослідити даний продукт на технічному рівні.
3. Зробити висновки, щодо даного продукту.

**Беклог проекту із розбитими під задачами:**

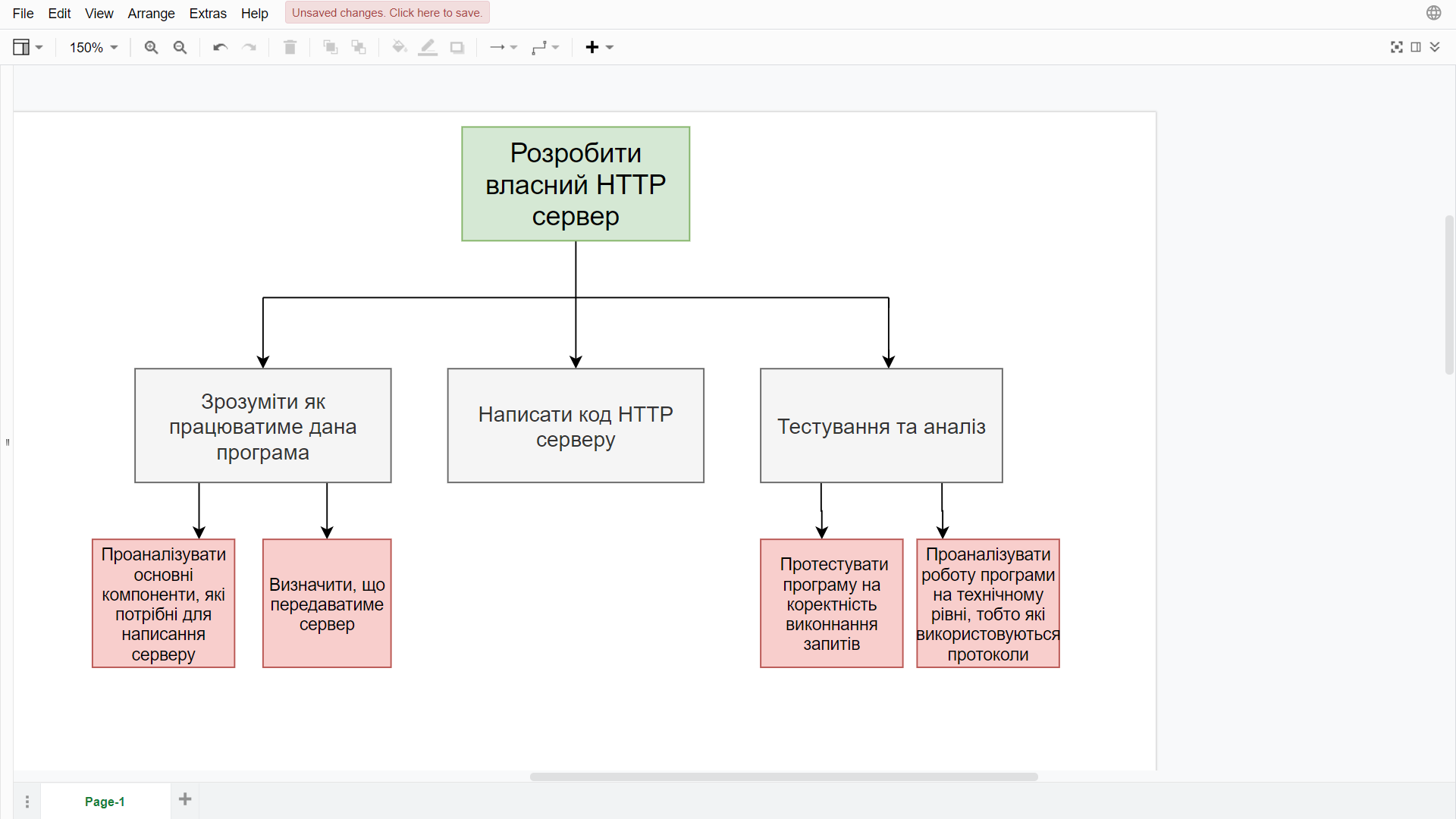
1. **Реалізація програми (HTTP серверу):**
2. Написати функцію доступу до підключення клієнта;
3. Створити клас, який допомагатиме для вводу та виводу даних;
4. Написати метод, який виводитиме повідомлення на екран;
5. Написати метод, який зчитуватиме повідомлення;
6. **Дослідити які протоколи передаються:**
7. Розглянути детально, запити які надсилаються серверу;
8. Розглянути відповіді, що надсилаються сервером;
9. Проглянути які протоколи передаються.
10. **Оформити документацію до проекту.**

**Хід виконання проекту**

1. Roadmap



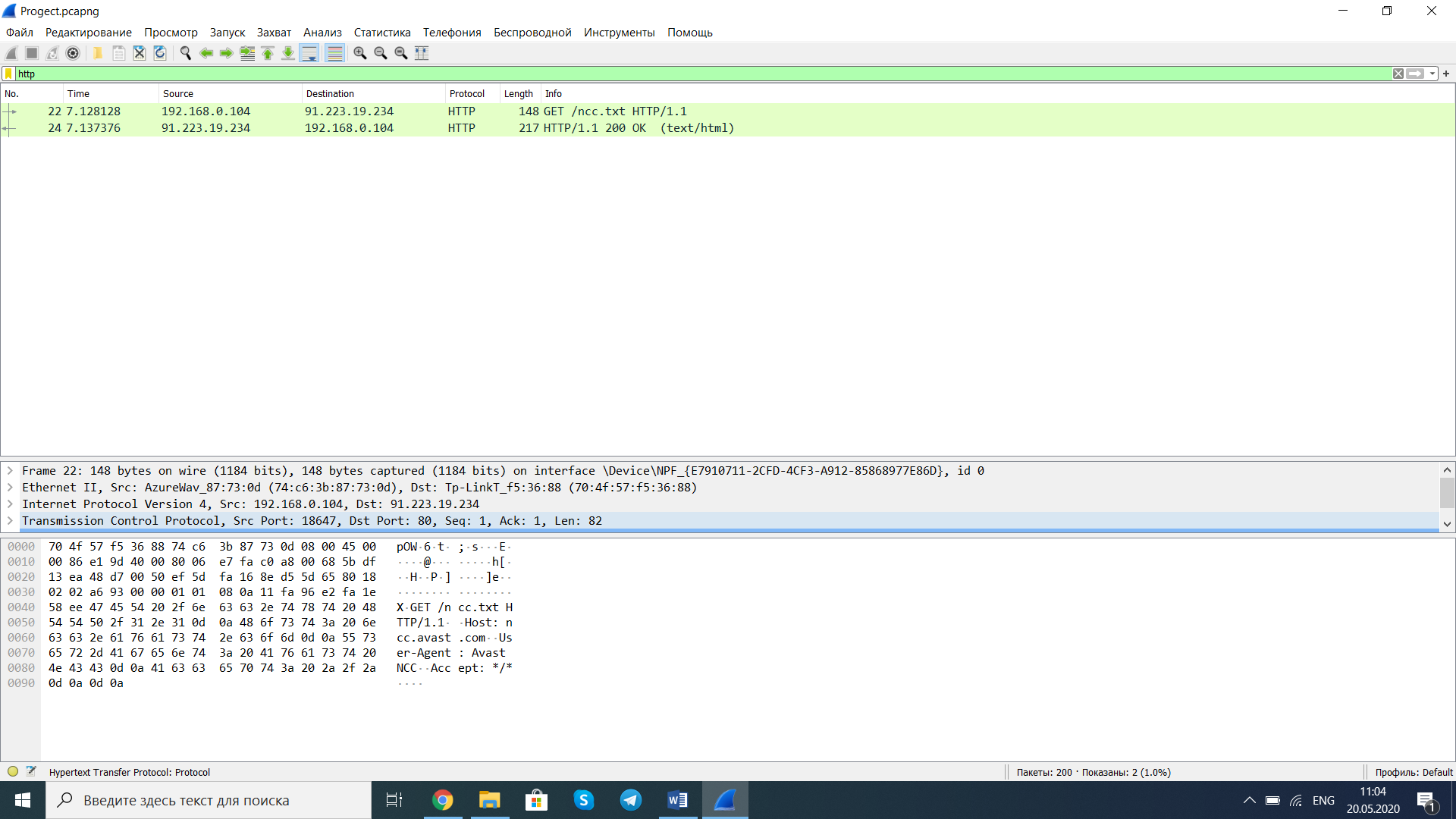
1. Декомпозиція проекту



1. Реалізація
   1. Код програми:

**package** com.kpi.http;  
  
**import** java.net.ServerSocket;  
**import** java.net.Socket;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
**import** java.io.InputStreamReader;  
**import** java.io.BufferedReader;  
  
*/\*\*  
 \* The type Http server.  
 \*/***public class** HttpServer {  
  
 **private static class** SocketProcessor **implements** Runnable {  
  
 */\*\*  
 \* The S.  
 \*/* **public** Socket **s**;  
 */\*\*  
 \* The Is.  
 \*/* **public** InputStream **is**;  
 */\*\*  
 \* The Os.  
 \*/* **public** OutputStream **os**;  
  
 */\*\*  
 \* Instantiates a new Socket processor.  
 \*  
 \** ***@param s*** *the s  
 \** ***@throws*** *Throwable the throwable  
 \*/* **public** SocketProcessor(Socket s) **throws** Throwable {  
 **this**.**s** = s;  
 **this**.**is** = s.getInputStream();  
 **this**.**os** = s.getOutputStream();  
 }  
  
 **public void** run() {  
 **try** {  
 readInputHeaders();  
 writeResponse(**"<html><body><h1>The project done: IASA 2020</h1></body></html>"**);  
 } **catch** (Throwable t) {  
 } **finally** {  
 **try** {  
 **s**.close();  
 } **catch** (Throwable t) {  
  
 }  
 }  
 System.***err***.println(**"Client processing finished"**);  
 }  
  
 */\*\*  
 \* Write response.  
 \*  
 \** ***@param s*** *the s  
 \** ***@throws*** *Throwable the throwable  
 \*/* **public void** writeResponse(String s) **throws** Throwable {  
 String response = **"HTTP/1.1 200 OK\r\n"** +  
 **"Server: YarServer/2009-09-09\r\n"** +  
 **"Content-Type: text/html\r\n"** +  
 **"Content-Length: "** + s.length() + **"\r\n"** +  
 **"Connection: close\r\n\r\n"**;  
 String result = response + s;  
 **os**.write(result.getBytes());  
 **os**.flush();  
 }  
  
 */\*\*  
 \* Read input headers.  
 \*  
 \** ***@throws*** *Throwable the throwable  
 \*/* **public void** readInputHeaders() **throws** Throwable {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(**is**));  
 **while**(**true**) {  
 String s = br.readLine();  
 **if**(s == **null** || s.trim().length() == 0) {  
 **break**;  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* The entry point of application.  
 \*  
 \** ***@param args*** *the input arguments  
 \** ***@throws*** *Throwable the throwable  
 \*/* **public static void** main(String[] args) **throws** Throwable {  
 ServerSocket ss = **new** ServerSocket(8080);  
 **while** (**true**) {  
 Socket s = ss.accept();  
 System.***err***.println(**"Client accepted"**);  
 **new** Thread(**new** SocketProcessor(s)).start();  
 }  
 }  
}

* 1. Опис протоколів, що передаються

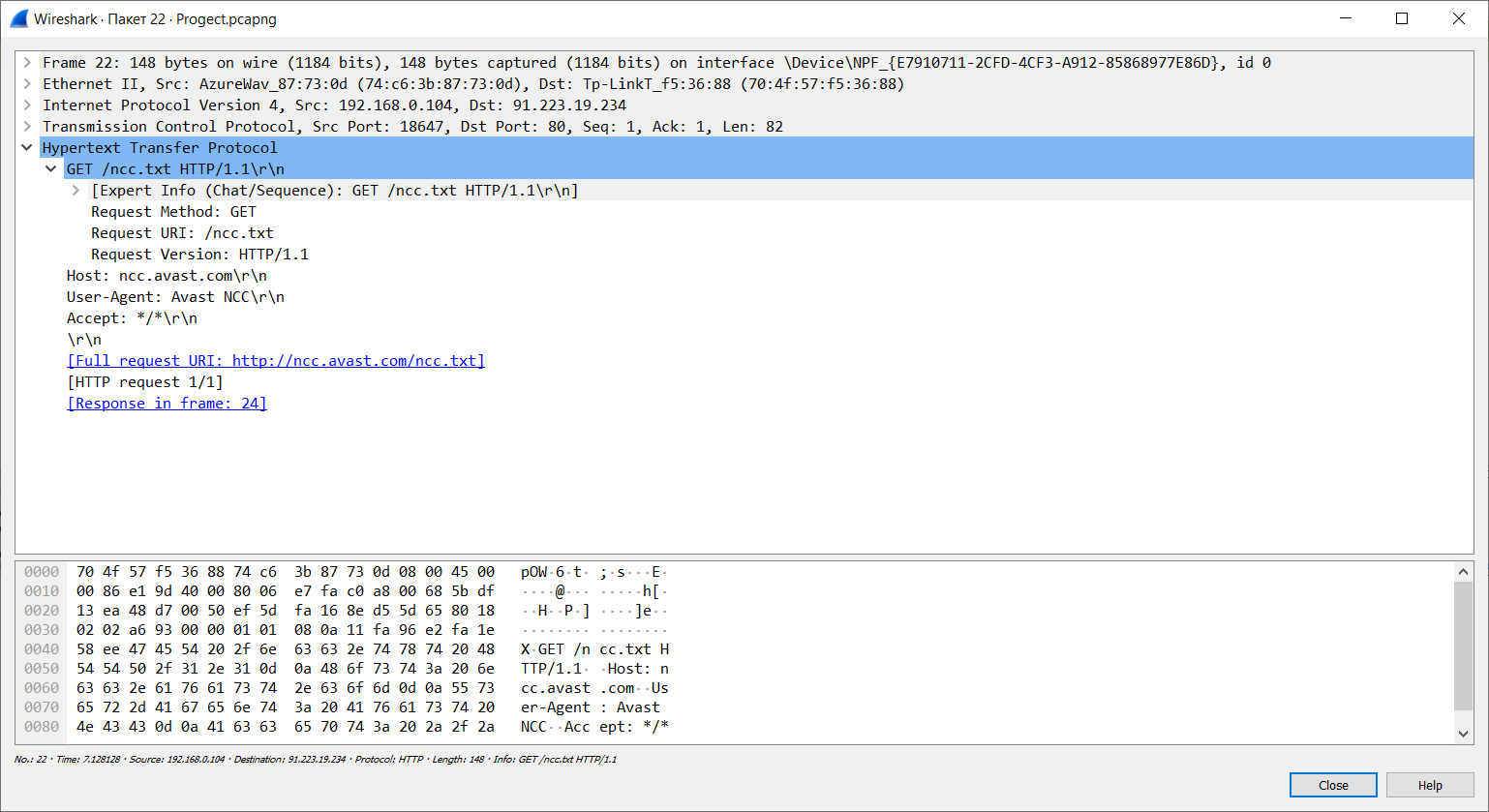


Протокол HTTP:

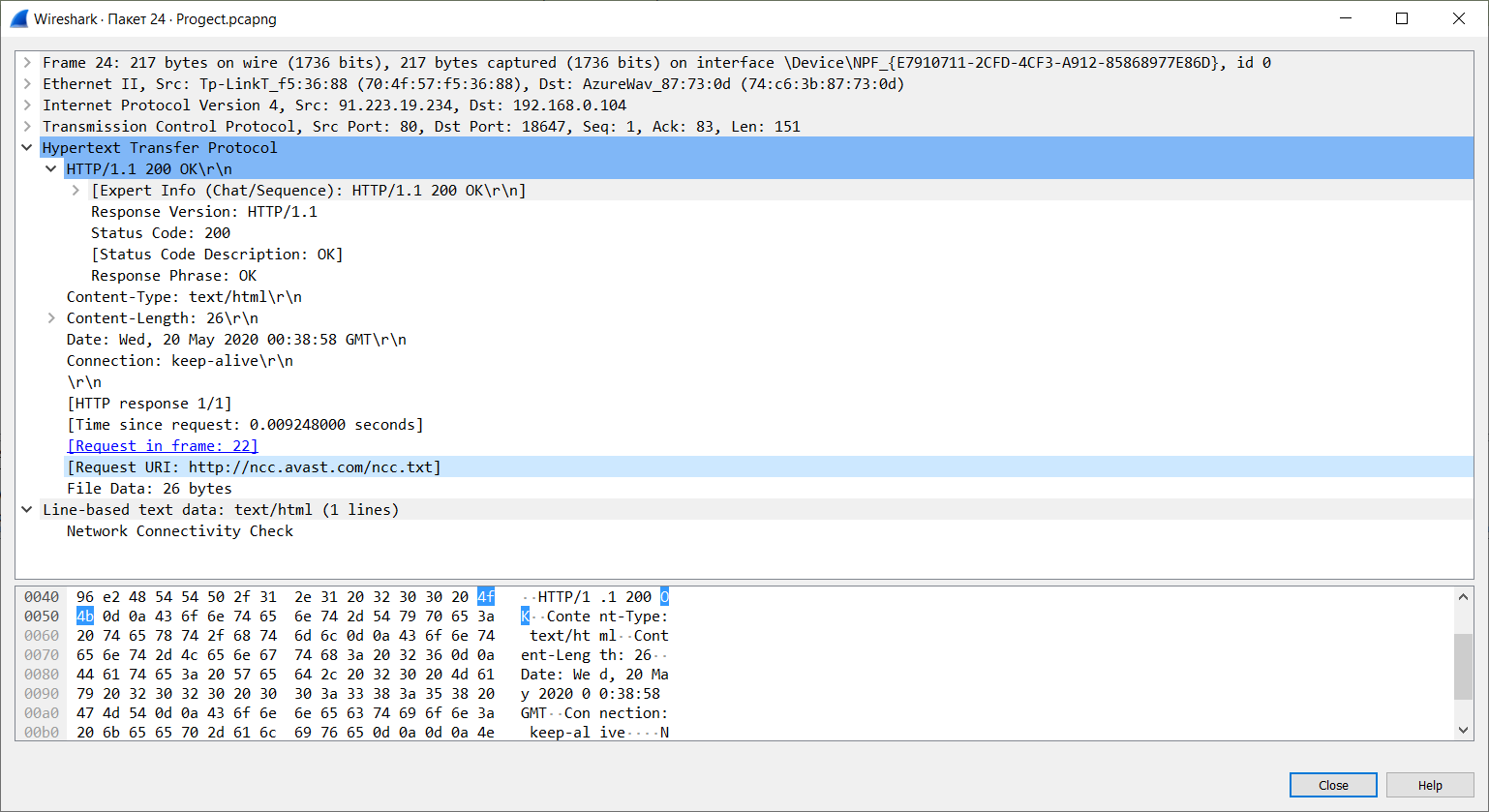
Час за який сервер обробляє запит та надсилає відповідь це

0,009 248‬ с.

Запит, який подається серверу :

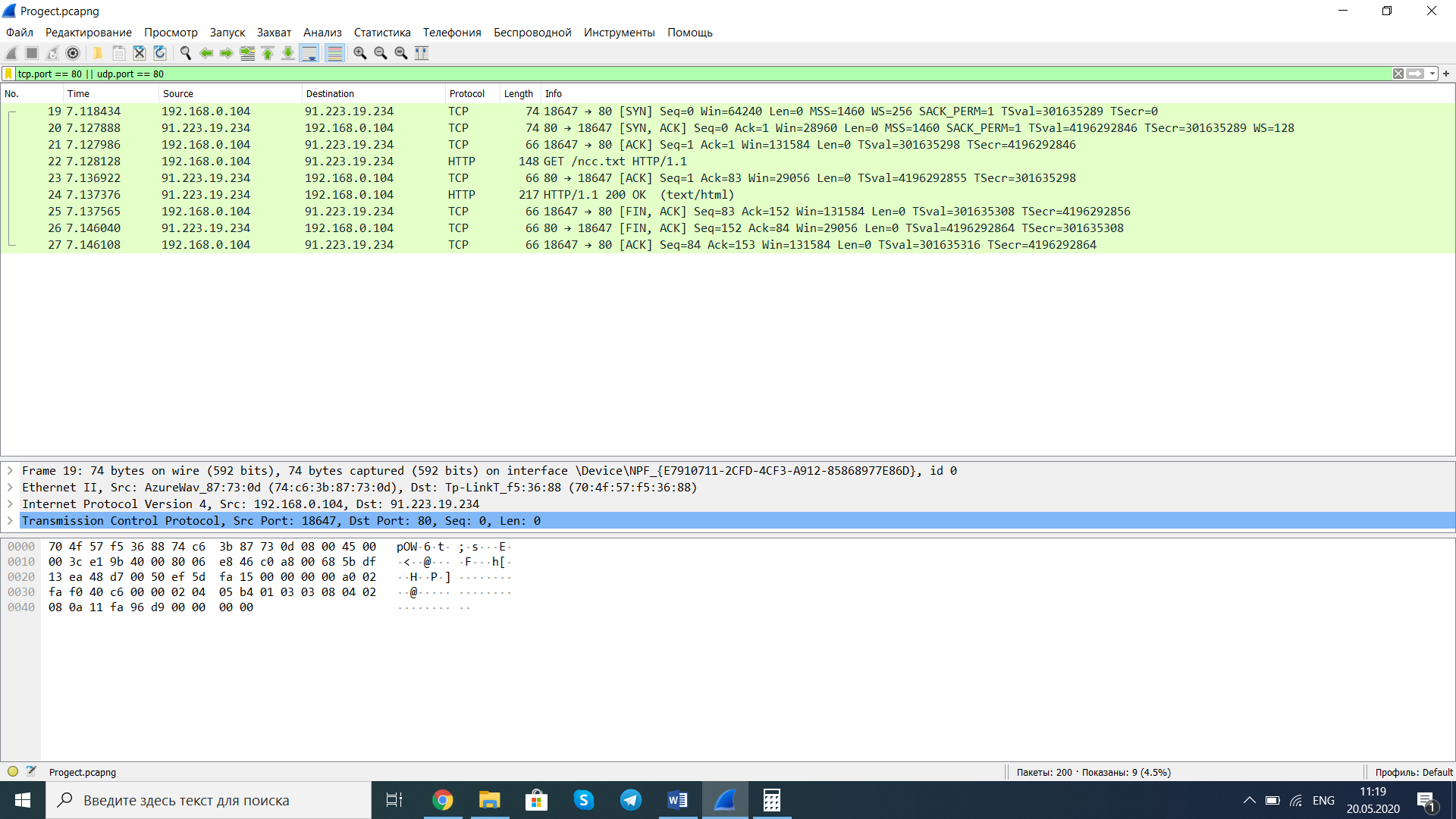


Відповідь сервера:



Можна замітити, що запит був коректно оброблено та була надана відповідь для користувача.

Окрім протоколів HTTP передаються також протоколи TCP



**Висновок:** Під час виконання даного проекту, було покращено навички роботи з Java та роботи з протоколами. Було продемонстровано протоколи, які передаються при розробці власного проекту, який пов’язаний з мережевим передаванням даних. У результати виконання роботи, програма коректно працює, приймає запити та надає відповіді клієнту.