

Перший етап виконання курсової роботи

1. Визначення та обґрунтування теми дослідження.

Освоєння основ по використанню API для отримання даних з мережі Інтернет.

Затвердження теми курсової роботи.

	Завдання	Результат виконання завдання
1.1	Визначити та описати проблему, вирішенню якої буде присвячена курсова робота.	1 сторінка А4, 10 шрифт
1.2	Ознайомитися з короткими нотатками та обрати API з вказаного переліку.	
1.3	Ознайомитися з документацією по обраному API.	
1.4	Ознайомитися з документацією Python оболонки для роботи з обраним API.	
1.5	Встановити необхідне програмне забезпечення (Python модулі та залежні бібліотеки). Перевірити працездатність програми оболонки.	
1.6	Написати короткий опис функціональних можливостей API.	1 сторінка А4, 10 шрифт
1.7	Навести приклад використання вказаного API за допомогою програми оболонки.	1 сторінка А4, 10 шрифт Python модуль
1.8	На основі функціональних можливостей API розробити пропозицію по проведенню дослідження на основі даних, які дозволяє отримати API. Розробити пропозицію допоможуть відповіді на наступні питання: Які дані я можу зібрати? Що я хочу отримати після обробки цих даних?. Яку користь можуть мати результати обробки? Пропозицію по проведенню дослідження оформити у вигляді вимоги на систему.	1 сторінка А4, 10 шрифт
1.9	Затвердити тему курсової роботи.	1 сторінка А4, згідно з наданим шаблоном

Короткі нотатки до першого етапу виконання курсової роботи

Перший етап виконання курсової роботи передбачає вирішення трьох основних завдань

1. Визначення тематики дослідження, якому буде присвячена курсова робота.
2. Пошук та визначення засобів для отримання необхідних даних для проведення дослідження.
3. Затвердження теми курсової роботи.

1. Тематика дослідження, якому буде присвячена курсова роботи повинна

викликати зацікавлення у виконавця, але мати реальні перспективи по реалізації в межах курсової роботи. Потрібно спробувати описати проблему, яку може допомогти вирішити проведення дослідження. Якщо проблема зрозуміла то тоді можна спробувати описати програму, яку потрібно розробити щоб виконати дослідження. Опис програми просто зробити наступним чином. На листку паперу зверху написати назву програми, а далі спробувати написати все про те якою бачиться ця програма. Якщо вдалося написати такий опис обсягом хоча б у 100 слів то варто прочитати цей опис комусь ще наприклад, колезі та обговорити його з ним. Також доцільно поцікавитися чи такі програми вже хтось розробляв. Можливо вже хтось виконував аналогічні чи подібні дослідження. Якщо це так то не потрібно відмовлятися від своєї ідеї, але потрібно над нею ще подумати й знайти та описати те що буде вирізняти ваше дослідження і вашу програму від тих аналогів чи прототипів, які ви знайшли.

Далі потрібно розробити вимогу на систему. Система це і є та програма яку буде розроблятися. Вимога на систему це перший документ, який обов'язково розробляють перед початком будь яких робіт по розробці програмного продукту (програми, програмного забезпечення). Цей документ повинен дати відповідь на питання чи доцільно розробляти програмний продукт, іншими словами в ньому дається опис бізнес причин для його створення та вказується, яку користь можна очікувати від цього продукту. Після створення вимоги на систему її затверджує група осіб або одна керівна відповідальна особа.

В наступній таблиці наведено основні складові частини цього документу та приклади, як можна описати ці складові частини з коментарями.

Складові частини вимоги на систему (Systems analysis and design /Alan Dennis, Barbara Haley Wixom, Roberta M. Roth.-5th ed. 2012)

	Складова частина	Опис	Приклад	Коментар
1	Спонсор проекту (Project Sponsor)	Особа або група осіб які є ініціаторами проекту. Це особи з якими в першу чергу потрібно буде обговорювати бізнес-сторону проекту.	Працівники фінансового відділу. Віце-президент. ІТ менеджер. CIO. CEO.	У випадку курсової роботи достатньо вказати себе як ініціатора проекту
2	Бізнес потреба (Business Need)	Бізнес причини що спонукали до ініціації проекту (системи, програмного продукту).	Збільшити продажі. Збільшити частку ринку. Покращити доступ до інформації. Покращити обслуговування клієнтів. Виправити дефекти продукту. Спростити процеси придбання продукту.	
3	Бізнес вимоги (Business Requirements)	Бізнес можливості, які буде забезпечувати система.	Зберігає демографічні дані про користувачів. Дозволяє створювати управлінські звіти. Забезпечує online підтримку користувачів. Надає online доступ до інформації. Підтримує можливість пошуку продукту.	Тут не потрібно вказувати як буде працювати програма, а тільки те що буде основними результатами її роботи.
4	Бізнес вигоди (користь) (Business Value)	Вигоди, які буде набуто після створення системи	Збільшення продажів на 3%. 1% збільшення частки ринку. Зменшення кількості працівників на 5*(мінімальна з.п.)	Цей пункт вимог на систему можна опустити, але подумати наскільки

			200 000 економії від зменшення витрат на постачання. 150 000 економії внаслідок заміни теперішньої системи	корисною може бути ваш продукт все ж доцільно.
5	Питання та обмеження (Special Issues or Constraints)	Питання, які стосуються впровадження системи і які повинні бути відомі групі осіб або одній особі, яка затверджує вимогу на систему	Граничний термін розробки – 25 травня. Команда розробників повинна отримати вищий рівень доступу для роботи з даними.	

Після розробки вимоги на систему потрібно повернутися до початкового опису і якщо потрібно внести в нього зміни.

2. Дані з мережі Інтернет можна переглядати або завантажувати за допомогою програм браузерів. Підприємства, які володіють ресурсами в Інтернеті також надають і web API (Application Program Interface). Web API можна вважати сервісом, який надає підприємство для доступу до своїх даних, які розміщені в мережі Інтернет. Для отримання цих даних потрібно надіслати до API відповідний запит.

Компанія Google надає різноманітні сервіси, наприклад, пошук відстані між двома точками, визначення висоти над рівнем моря тощо. Для забезпечення надання таких сервісів постачальник API має базовий URL, на основі якого будується запит. У випадку складних API після базового URL вказують також і кінцеву точку (endpoint, endpoints – дозволяють організувати доступ до різних даних). Наприклад, API Google Distance Matrix має базовий URL <https://maps.googleapis.com/maps/api/distancematrix/json?>, а API Google Maps Elevation API - <https://maps.googleapis.com/maps/api/elevation/json?>. Якщо перейти в браузері за першим URL то на екрані будуть зображатися наступне:

```
{
  "destination_addresses" : [],
  "origin_addresses" : [],
  "rows" : [],
  "status" : "INVALID_REQUEST"
}
```

Будь які дані у відповіді відсутні. Для того щоб отримати певні дані API повинен знати, які дані надати у відповідь на запит. Для цього в запиті разом з базовим URL потрібно передати параметри даних, які необхідно отримати. Ці параметри записуються у вигляді пари "ключ = значення" після символу "?". Наприклад, для Distance Matrix API, такими парами можуть бути "origins=Городок" та "destinations=Львів" (для Elevation API locations= 48.5000, 23.2703), і тоді URL буде наступний: "https://maps.googleapis.com/maps/api/distancematrix/json?origins=Городок&destinations=Львів". Якщо перейти за цим URL то можна побачити дані, які повертає (надає) API.

```
{
  "destination_addresses" : [ "Львів, Львівська область, Україна" ],
  "origin_addresses" : [ "Городок, Львівська область, Україна" ],
  "rows" : [
    {
      "elements" : [
        {
          "distance" : {
            "text" : "28,8 км",
            "value" : 28792
          },
          "duration" : {
            "text" : "44 хв",
            "value" : 2617
          }
        }
      ]
    }
  ]
}
```

```

        "status" : "OK"
    }
]
},
"status" : "OK"
}

```

Для використання API у власному коді, потрібно реалізувати запит (request) засобами Python та вилучити (отримати) дані. Реалізувати запит у власному коді дозволяє пакунок стандартної бібліотеки `urllib`. Модуль `urllib.request` з цього пакунку дозволяє зробити такий запит та отримати дані. Дані будуть отримані у двійковому вигляді й потрібно скористатися відповідним паунком зі стандартної бібліотеки (в залежності від формату даних `json` або `xml`) для перетворення цих даних у потрібний вигляд (наприклад, словник).

Для реалізації всього вище переліченого можна написати, наприклад наступну функцію. Функція `get_data_from_URL` (`base_url`, `parameters`), яка буде приймати, базовий URL та словник параметрів і повертати дані.

Варіант програми для отримання даних у форматі XML

```

#Function get_data_from_URL takes in a URL and parameters
#for an API request and then returns the data (XML).
#Parameters must be passed in as a dictionary of key value pairs.

import urllib.request
import urllib.parse
import xml.etree.ElementTree as ET

BASE_URL = "https://maps.googleapis.com/maps/api/elevation/xml?"
params = { 'locations': "48.5000,23.2703" }

def get_data_from_URL(base_url, options):

    params_str = urllib.parse.urlencode(params)
    request_url = base_url + params_str
    request = urllib.request.Request(request_url)
    with urllib.request.urlopen(request_url) as response:
        # read the response from the response object
        data = response.read()
        # decode it from bytes to string
        data = data.decode("utf-8")
        # parse the string into a element tree
        root = ET.fromstring(data)
    return root

data = get_data_from_URL(BASE_URL, params)

for element in data:
    print(element.tag, element.getchildren())

```

Варіант програми для отримання даних у форматі json

```
#Function get_data_from_URL takes in a URL and parameters
#for an API request and then returns the data (json).
#Parameters must be passed in as a dictionary of key value pairs.

import urllib.request
import urllib.parse
import json

BASE_URL = "https://maps.googleapis.com/maps/api/elevation/json?"
params = { 'locations': "48.5000,23.2703"}

def get_data_from_URL(base_url, options):

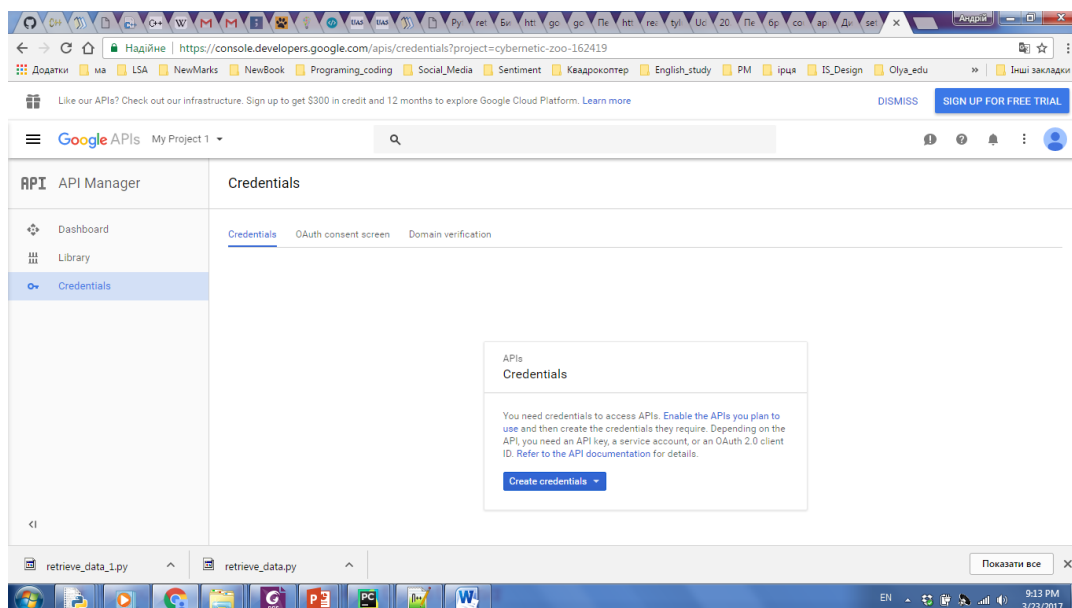
    params_str = urllib.parse.urlencode(params)
    request_url = base_url + params_str
    request = urllib.request.Request(request_url)
    with urllib.request.urlopen(request_url) as response:
        # read the response from the response object
        data = response.read()
        # decode it from bytes to string
        data = data.decode("utf-8")
        # parse the string into a python object of dictionaries and lists
        json.loads(data)
    return data

data = get_data_from_URL(BASE_URL, params)

print(data)
```

Для того щоб отримати можливість використовувати інші Google API або API інших організацій потрібно виконати вказану в документації до API послідовність кроків по реєстрації на отримання API Key. В різних постачальників API ця послідовність кроків може відрізнятися. Для отримання API Key в Google API потрібно:

1. На сторінці Google Developer Console (<https://console.developers.google.com>) та створити новий проект.
2. За допомогою панелі "API Manager" зліва вибрати пункт "Credentials"
3. В закладці "OAuth consent screen" заповнити необхідні поля.
4. В закладці "Credentials" вибрати "Create credentials" -> "API Key" та створити API Key .



Деякі постачальники API надають дані без необхідності отримання API Key. Наприклад, Udacity.

Для спрощення формування, надсилання та обробки результатів запитів розробляються спеціальні програми оболонки (python-api-wrappers). За наступним посиланням (<https://github.com/realpython/list-of-python-api-wrappers>) можна побачити великий перелік таких програм, які розроблені мовою програмування Python. Для того щоб мати можливість скористатися можливостями цих програм потрібно встановити ці пакунки модулів згідно з вказівками у їх документації. Потрібно зазначити що кожна така програма потребує перевірки на працездатність та коректність роботи. Тому рішення про використання тої чи іншої програми потрібно приймати після проведення такої перевірки.