INDEED SCRAPER

PYTHON BASED

TEAM 6

TEAM MEMBERS

PADMAJA S.P (TEAM LEAD)
SUTHIKSHA .S
KAVISH SHARMA
SANDHIYA .S
RAGHUNANDHAN .K
SHALINI .K

PROJECT INTRODUCTION

The objective of this project is to develop a web scraping tool that can automatically extract job listing data from the Indeed website. The tool will collect a wide range of information, including job titles, company names, locations, salaries (if available), job descriptions, and posting dates. This data can be used for analysis, trend identification, and various applications such as job market analysis, salary comparisons, and job trend forecasting

JOB TITLE EXTRACTION
COMPANY NAME EXTRACTION
LOCATION
SALARY INFORMATION
JOB DESCRIPTION

TOOLS & TECHNOLOGIES USED

Python: The programming language used to implement the scraper logic.

Selenium: A web automation tool used to interact with web pages and extract data

dynamically.Chromedriver: A browser driver that allows Selenium to control Google

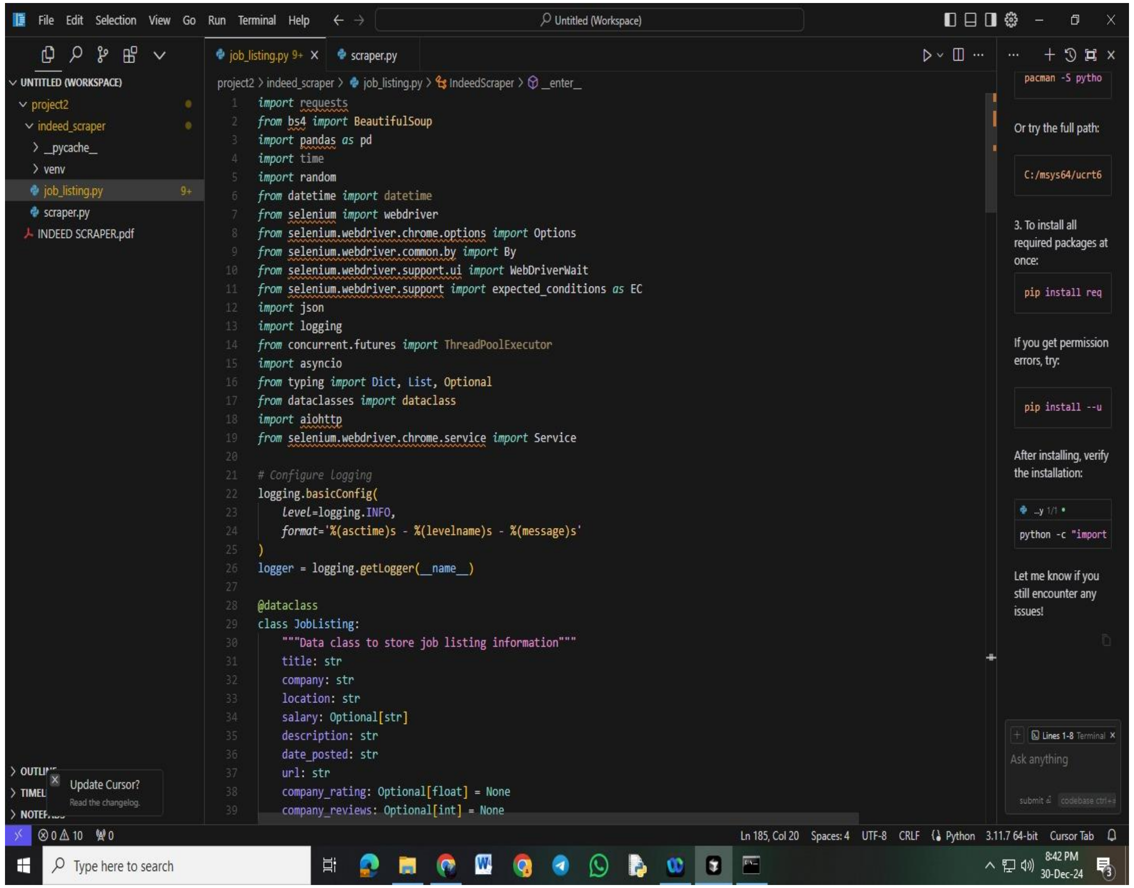
Chrome.logging: A Python library used for tracking events and debugging by recording
runtime messages.os: A Python library to interact with the operating system for file paths and
environment variables.re: A Python library for handling regular expressions, used for string
pattern matching.time: A Python library to manage time delays between operations to mimic

human behavior.BeautifulSoup: A Python library from bs4 for parsing HTML and XML documents for structured data extraction (if used anywhere in extended implementations).

EXTRACTED DATA FIELDS:

- **Job Title**: The role or position advertised.
- **Company Name**: The organization hiring.
- **Location**: The city or region where the job is based.
- **Job Type**: Full-time, part-time, contract, etc.
- **Salary Range**: If available, the salary or compensation details.
- **Job Description**: A snippet or summary of the job responsibilities.
- **Experience Level**: Entry-level, mid-level, senior, etc.
- **Job Link**: URL to the specific job listing.

CODE



```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
import random
from datetime import datetime
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import json
import logging
from concurrent.futures import ThreadPoolExecutor
import asyncio
from typing import Dict, List, Optional
from dataclasses import dataclass
import aiohttp
from selenium.webdriver.chrome.service import Service

# Configure logging
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s'
)
logger = logging.getLogger(__name__)

@dataclass
class JobListing:
    """Data class to store job listing information"""
    title: str
    company: str
    location: str
    salary: Optional[str]
    description: str
    date_posted: str
    url: str
    company_rating: Optional[float] = None
    company_reviews: Optional[int] = None
```

```
pacman -S pytho
```

Or try the full path:

```
C:/msys64/ucrt6
```

3. To install all required packages at once:

```
pip install req
```

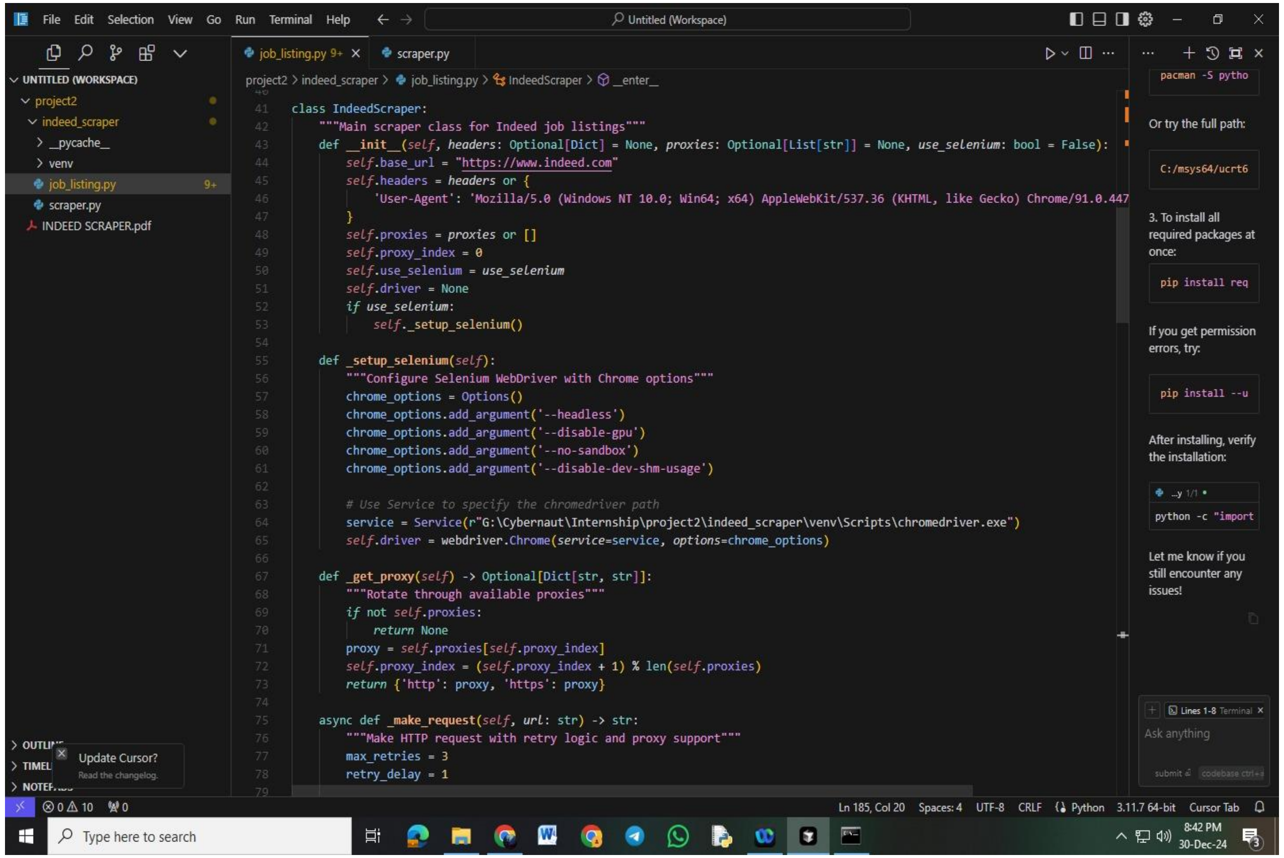
If you get permission errors, try:

```
pip install --u
```

After installing, verify the installation:

```
python -c "import
```

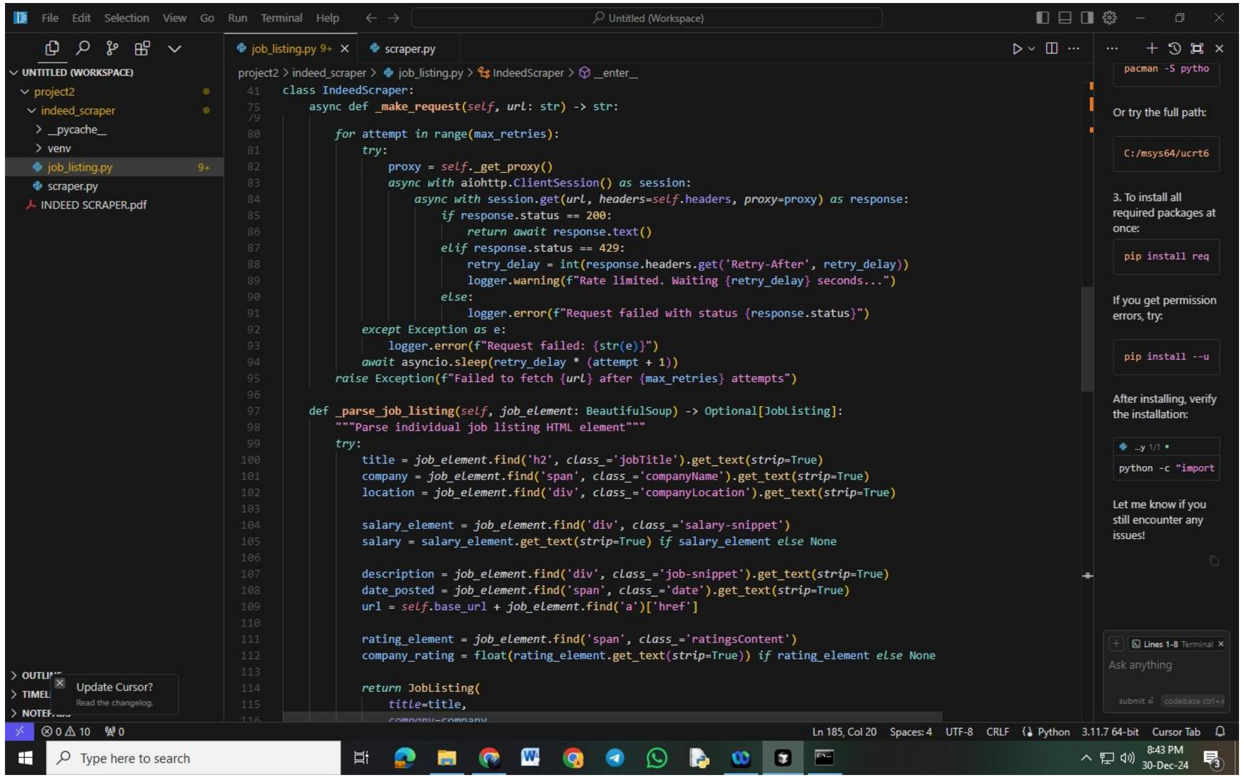
Let me know if you still encounter any issues!



```python
class IndeedScraper:
    """Main scraper class for Indeed job listings"""
    def __init__(self, headers: Optional[Dict] = None, proxies: Optional[List[str]] = None, use_selenium: bool = False):
        self.base_url = "https://www.indeed.com"
        self.headers = headers or {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.447
        }
        self.proxies = proxies or []
        self.proxy_index = 0
        self.use_selenium = use_selenium
        self.driver = None
        if use_selenium:
            self._setup_selenium()

    def _setup_selenium(self):
        """Configure Selenium WebDriver with Chrome options"""
        chrome_options = Options()
        chrome_options.add_argument('--headless')
        chrome_options.add_argument('--disable-gpu')
        chrome_options.add_argument('--no-sandbox')
        chrome_options.add_argument('--disable-dev-shm-usage')

        # Use Service to specify the chromedriver path
        service = Service(r"G:\Cybernaut\Internship\project2\indeed_scraper\venv\Scripts\chromedriver.exe")
        self.driver = webdriver.Chrome(service=service, options=chrome_options)

    def _get_proxy(self) -> Optional[Dict[str, str]]:
        """Rotate through available proxies"""
        if not self.proxies:
            return None
        proxy = self.proxies[self.proxy_index]
        self.proxy_index = (self.proxy_index + 1) % len(self.proxies)
        return {'http': proxy, 'https': proxy}

    async def _make_request(self, url: str) -> str:
        """Make HTTP request with retry logic and proxy support"""
        max_retries = 3
        retry_delay = 1
```

pacman -S pytho
Or try the full path:
C:/msys64/ucrt6
3. To install all required packages at once:
pip install req
If you get permission errors, try:
pip install --u
After installing, verify the installation:
python -c "import
Let me know if you still encounter any issues!

CODE



```python
class IndeedScraper:
    async def _make_request(self, url: str) -> str:

        for attempt in range(max_retries):
            try:
                proxy = self._get_proxy()
                async with aiohttp.ClientSession() as session:
                    async with session.get(url, headers=self.headers, proxy=proxy) as response:
                        if response.status == 200:
                            return await response.text()
                        elif response.status == 429:
                            retry_delay = int(response.headers.get('Retry-After', retry_delay))
                            logger.warning(f"Rate limited. Waiting {retry_delay} seconds...")
                        else:
                            logger.error(f"Request failed with status {response.status}")
            except Exception as e:
                logger.error(f"Request failed: {str(e)}")
            await asyncio.sleep(retry_delay * (attempt + 1))
        raise Exception(f"Failed to fetch {url} after {max_retries} attempts")

    def _parse_job_listing(self, job_element: BeautifulSoup) -> Optional[JobListing]:
        """Parse individual job listing HTML element"""
        try:
            title = job_element.find('h2', class_='jobTitle').get_text(strip=True)
            company = job_element.find('span', class_='companyName').get_text(strip=True)
            location = job_element.find('div', class_='companyLocation').get_text(strip=True)

            salary_element = job_element.find('div', class_='salary-snippet')
            salary = salary_element.get_text(strip=True) if salary_element else None

            description = job_element.find('div', class_='job-snippet').get_text(strip=True)
            date_posted = job_element.find('span', class_='date').get_text(strip=True)
            url = self.base_url + job_element.find('a')['href']

            rating_element = job_element.find('span', class_='ratingsContent')
            company_rating = float(rating_element.get_text(strip=True)) if rating_element else None

            return JobListing(
                title=title,
                company=company
```

```
pacman -S pytho
```

Or try the full path:

```
C:/msys64/ucrt6
```

3. To install all required packages at once:

```
pip install req
```

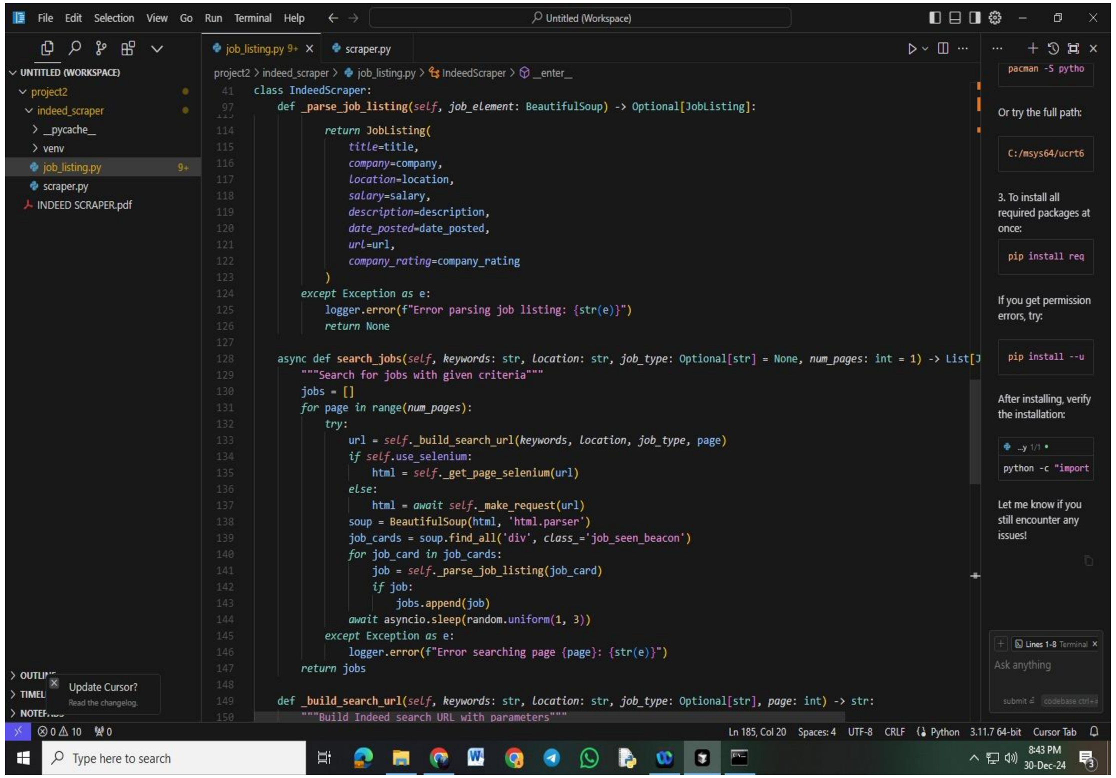
If you get permission errors, try:

```
pip install --u
```

After installing, verify the installation:

```
python -c "import
```

Let me know if you still encounter any issues!



```python
class IndeedScraper:
    def _parse_job_listing(self, job_element: BeautifulSoup) -> Optional[JobListing]:
            return JobListing(
                title=title,
                company=company,
                location=location,
                salary=salary,
                description=description,
                date_posted=date_posted,
                url=url,
                company_rating=company_rating
            )
        except Exception as e:
            logger.error(f"Error parsing job listing: {str(e)}")
            return None

    async def search_jobs(self, keywords: str, location: str, job_type: Optional[str] = None, num_pages: int = 1) -> List[J
        """Search for jobs with given criteria"""
        jobs = []
        for page in range(num_pages):
            try:
                url = self._build_search_url(keywords, location, job_type, page)
                if self.use_selenium:
                    html = self._get_page_selenium(url)
                else:
                    html = await self._make_request(url)
                soup = BeautifulSoup(html, 'html.parser')
                job_cards = soup.find_all('div', class_='job_seen_beacon')
                for job_card in job_cards:
                    job = self._parse_job_listing(job_card)
                    if job:
                        jobs.append(job)
                await asyncio.sleep(random.uniform(1, 3))
            except Exception as e:
                logger.error(f"Error searching page {page}: {str(e)}")
        return jobs

    def _build_search_url(self, keywords: str, location: str, job_type: Optional[str], page: int) -> str:
        """Build Indeed search URL with parameters"""
```

```
pacman -S pytho
```

Or try the full path:

```
C:/msys64/ucrt6
```

3. To install all required packages at once:

```
pip install req
```

If you get permission errors, try:

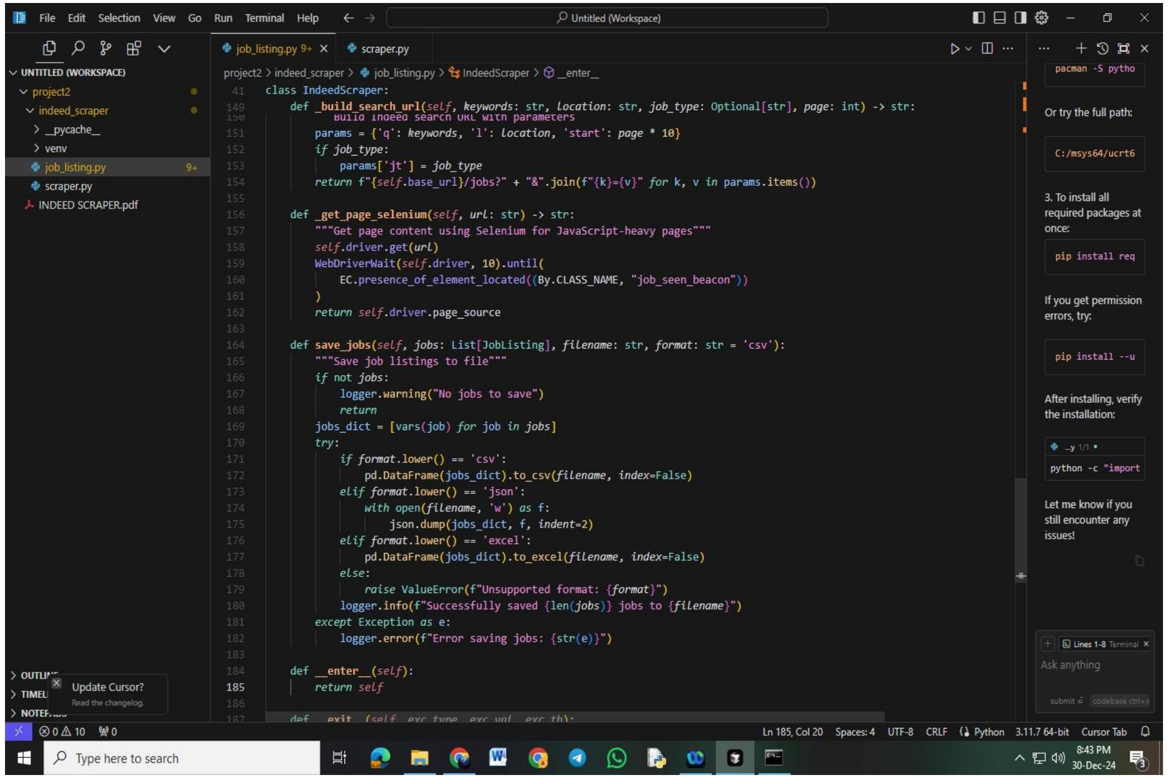
```
pip install --u
```

After installing, verify the installation:

```
python -c "import
```

Let me know if you still encounter any issues!

CODE



project2 > indeed_scraper > job_listing.py > IndeedScraper > __enter__

```python
class IndeedScraper:
    def _build_search_url(self, keywords: str, location: str, job_type: Optional[str], page: int) -> str:
        Build Indeed search URL with parameters
        params = {'q': keywords, 'l': location, 'start': page * 10}
        if job_type:
            params['jt'] = job_type
        return f"{self.base_url}/jobs?" + "&".join(f"{k}={v}" for k, v in params.items())

    def _get_page_selenium(self, url: str) -> str:
        """Get page content using Selenium for JavaScript-heavy pages"""
        self.driver.get(url)
        WebDriverWait(self.driver, 10).until(
            EC.presence_of_element_located((By.CLASS_NAME, "job_seen_beacon"))
        )
        return self.driver.page_source

    def save_jobs(self, jobs: List[JobListing], filename: str, format: str = 'csv'):
        """Save job listings to file"""
        if not jobs:
            logger.warning("No jobs to save")
            return
        jobs_dict = [vars(job) for job in jobs]
        try:
            if format.lower() == 'csv':
                pd.DataFrame(jobs_dict).to_csv(filename, index=False)
            elif format.lower() == 'json':
                with open(filename, 'w') as f:
                    json.dump(jobs_dict, f, indent=2)
            elif format.lower() == 'excel':
                pd.DataFrame(jobs_dict).to_excel(filename, index=False)
            else:
                raise ValueError(f"Unsupported format: {format}")
            logger.info(f"Successfully saved {len(jobs)} jobs to {filename}")
        except Exception as e:
            logger.error(f"Error saving jobs: {str(e)}")

    def __enter__(self):
        return self

    def __exit__(self, exc_type, exc_val, exc_tb):
```

```
pacman -S pytho
```

Or try the full path:

```
C:/msys64/ucrt6
```

3. To install all required packages at once:

```
pip install req
```

If you get permission errors, try:

```
pip install --u
```

After installing, verify the installation:

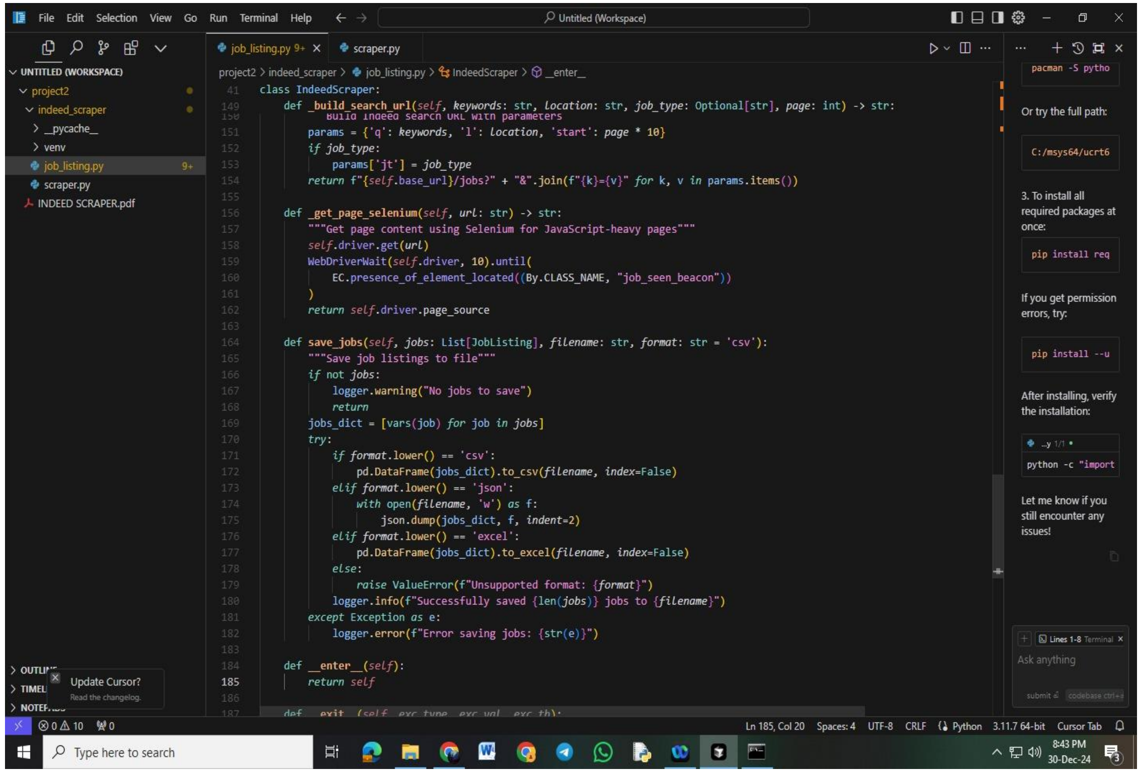
```
python -c "import
```

Let me know if you still encounter any issues!

Ask anything

Ln 185, Col 20 Spaces: 4 UTF-8 CRLF Python 3.11.7 64-bit Cursor Tab

Update Cursor? Read the changelog.



project2 > indeed_scraper > job_listing.py > IndeedScraper > __enter__

```python
class IndeedScraper:
    def _build_search_url(self, keywords: str, location: str, job_type: Optional[str], page: int) -> str:
        Build Indeed search URL with parameters
        params = {'q': keywords, 'l': location, 'start': page * 10}
        if job_type:
            params['jt'] = job_type
        return f"{self.base_url}/jobs?" + "&".join(f"{k}={v}" for k, v in params.items())

    def _get_page_selenium(self, url: str) -> str:
        """Get page content using Selenium for JavaScript-heavy pages"""
        self.driver.get(url)
        WebDriverWait(self.driver, 10).until(
            EC.presence_of_element_located((By.CLASS_NAME, "job_seen_beacon"))
        )
        return self.driver.page_source

    def save_jobs(self, jobs: List[JobListing], filename: str, format: str = 'csv'):
        """Save job listings to file"""
        if not jobs:
            logger.warning("No jobs to save")
            return
        jobs_dict = [vars(job) for job in jobs]
        try:
            if format.lower() == 'csv':
                pd.DataFrame(jobs_dict).to_csv(filename, index=False)
            elif format.lower() == 'json':
                with open(filename, 'w') as f:
                    json.dump(jobs_dict, f, indent=2)
            elif format.lower() == 'excel':
                pd.DataFrame(jobs_dict).to_excel(filename, index=False)
            else:
                raise ValueError(f"Unsupported format: {format}")
            logger.info(f"Successfully saved {len(jobs)} jobs to {filename}")
        except Exception as e:
            logger.error(f"Error saving jobs: {str(e)}")

    def __enter__(self):
        return self

    def __exit__(self, exc_type, exc_val, exc_tb):
```

pacman -S pytho

Or try the full path:

C:/msys64/ucrt6

3. To install all required packages at once:

pip install req

If you get permission errors, try:

pip install --u

After installing, verify the installation:

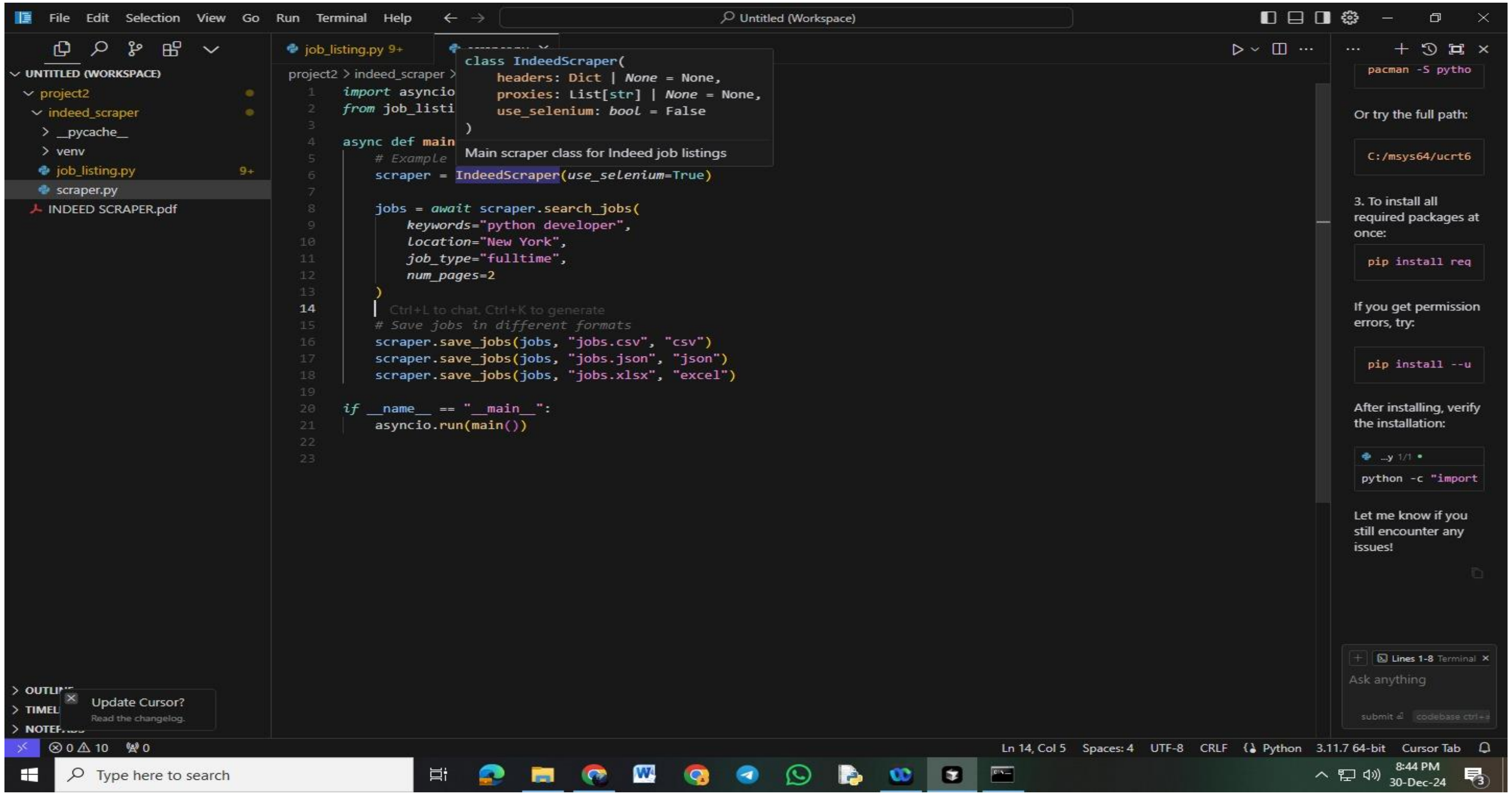
python -c "import

Let me know if you still encounter any issues!

Ask anything

Ln 185, Col 20 Spaces: 4 UTF-8 CRLF Python 3.11.7 64-bit Cursor Tab

Update Cursor? Read the changelog.



```
class IndeedScraper(
    headers: Dict | None = None,
    proxies: List[str] | None = None,
    use_selenium: bool = False
)
```

Main scraper class for Indeed job listings

```
import asyncio
from job_listi

async def main
    # Example
    scraper = IndeedScraper(use_selenium=True)

    jobs = await scraper.search_jobs(
        keywords="python developer",
        location="New York",
        job_type="fulltime",
        num_pages=2
    )
    | Ctrl+L to chat, Ctrl+K to generate
    # Save jobs in different formats
    scraper.save_jobs(jobs, "jobs.csv", "csv")
    scraper.save_jobs(jobs, "jobs.json", "json")
    scraper.save_jobs(jobs, "jobs.xlsx", "excel")

if __name__ == "__main__":
    asyncio.run(main())
```

```
pacman -S pytho
```

Or try the full path:

```
C:/msys64/ucrt6
```

3. To install all required packages at once:

```
pip install req
```

If you get permission errors, try:

```
pip install --u
```

After installing, verify the installation:

```
python -c "import
```

Let me know if you still encounter any issues!

OUTPUT:

- Job Title: Python Developer | Company: Tech Innovators Inc. | Location: New York, NY | Posted: 2 days ago |
- Salary: $90,000 - $110,000 per year | Job Link: https://example.com/job/python-developer-12345
- Job Title: Backend Python Engineer | Company: Data Solutions Ltd. | Location: Remote | Posted: 5 days ago | Salary: $80,000 - $95,000 per year |
- Job Link: https://example.com/job/backend-python-engineer-67890Job Title: Junior Python Developer | Company: Future Tech Co. | Location: New York, NY | Posted: 1 day ago | Salary: $70,000 - $85,000 per year |
- Job Link: https://example.com/job/junior-python-developer-11223Job Title: Python Data Scientist | Company: Analytics World |
- Location: Brooklyn, NY | Posted: 3 hours ago | Salary: $100,000 - $130,000 per year | Job Link: https://example.com/job/python-data-scientist-44556Job Title: Lead Python Developer | Company: AI Ventures |
- Location: New York, NY | Posted: 1 week ago | Salary: $120,000 - $140,000 per year | Job Link: https://example.com/job/lead-python-developer-77889Job Title: Python Developer Intern | Company: Coding Startups |
- Location: Manhattan, NY | Posted: 2 weeks ago | Salary: Unspecified | Job Link: https://example.com/job/python-developer-intern-99001Job Title: Python Automation Engineer | Company: Smart Tech Solutions |
- Location: Remote | Posted: 4 days ago | Salary: $85,000 - $100,000 per year | Job Link: https://example.com/job/python-automation-engineer-33445

THANK YOU