תיעוד הקוד עבור מטלת הבית – חלק 2:

מסמך זה מהווה תיעוד של האפליקציה שפיתחתי עבור מטלת הבית לחברת jeen-ai. במסמך זה פירטתי מה כל פונקציה עושה וכיצד היא משמשת אותנו בקוד.

.1

```python
def fetch_page_content(url):
    """Fetches and returns the HTML content of the given URL."""
    try:
        response = requests.get(url)
        response.raise_for_status()
        return response.text
    except requests.RequestException as e:
        print(f"Error fetching {url}: {e}")
        return None
```

פונקציה זו מקבלת כתובת URL ומבצעת request אל URL זה. אם הבקשה מצליחה היא מחזירה את תוכן ה-HTML של העמוד, במידה ומתרחשת שגיאה בבקשה (לדוגמה לא ניתן אישור לבצע request) הפונקציה תדפיס הודעת שגיאה ותחזיר None.

.2

```python
def parse_page_content(content):
    """Parses the HTML content of a page using BeautifulSoup."""
    soup = BeautifulSoup(content, "html.parser")
    title = soup.title.string if soup.title else "No Title"
    body_content = " ".join(p.get_text() for p in soup.find_all("p"))
    return title, body_content
```

פונקציה זו מקבלת את תוכן ה-HTML ומנתחת אותו באמצעות ספריית BeautifulSoup. הפונקציה מוצאת את הכותרת של העמוד (במידה ואין נקבל No Title) .

בנוסף, אנו מוצאים את כל הטקסט אשר קיים בתגיות <p> (הפסקאות שבעמוד).

הפונקציה מחזירה את הכותרת ואת התוכן שבתוך הפסקאות כטקסט מאוחד (title, body_content).

.3

```python
def extract_links(soup):
    """Extracts the links from a BeautifulSoup object."""
    links = [a["href"] for a in soup.find_all("a", href=True)]
    return links
```

פונקציה זו מקבלת אובייקט BeautifulSoup ומוצאת את כל הקישורים ע"י תגיות <a> עם תכונת ה-href שקיימים בעמוד.

הפונקציה מחזירה רשימה (links) של כל הקישורים אשר נמצאו בעמוד.

.4

```python
def crawl_website(url):
    """Crawls a website from a given URL and collects data."""
    visited = set()
    data = []
    counter = 0

    # Fetch content from the main page
    content = fetch_page_content(url)
    # If the content is None (Forbidden error like ksp.co.il), return the empty data -> empty excel file
    if content is None:
        return data

    # Parse the main page
    soup = BeautifulSoup(content, "html.parser")
    title, body_content = parse_page_content(content)
    data.append({"Page Title": title, "Page Url": url, "Page Content": body_content})

    # Extract links from the main page
    links = extract_links(soup)
```

```python
    # Fetch content from the extracted links
    for link in links:
        full_link = requests.compat.urljoin(url, link)
        if full_link not in visited and counter < 10 and full_link != url:
            content = fetch_page_content(full_link)
            if content is None:
                continue
            title, body_content = parse_page_content(content)
            data.append(
                {
                    "Page Title": title,
                    "Page Url": full_link,
                    "Page Content": body_content,
                }
            )
            counter += 1
            visited.add(full_link)

    return data
```

הפונקציה מבצעת סריקה של אתר אינטרנט החל מ-URL מסוים אשר יהיה בעל 2 רמות לפחות (לפי בקשת המטלה) ואוספת נתונים מהעמודים שנמצאים.

הפונקציה מתחילה מלבקר בעמוד הראשי, מוציאה את תוכן ה-HTML שלו ומנתחת את הכותרת ואת התוכן. לאחר מכן היא מוציאה את כל הקישורים מהעמוד הראשי, מבקרת בכל קישור (אני שמתי 10 קישורים כי יותר מזה יכול לקחת הרבה זמן, ניתן לבחור גם יותר קישורים) הפונקציה אוספת גם מהקישורים את הכותרת ואת התוכן וכל עמוד שביקרנו בו נשמר במערך ה data שנועד כדי להכיל את כל הקישורים שביקרנו בהם ובכך נוכל למנוע כפילויות.

.5

```
72  def save_data_to_excel(data, filename="web_crawl_data.xlsx"):
73      """Saves the collected data to an Excel file."""
74      df = pd.DataFrame(data)
75      df.to_excel(filename, index=False)
76      print(f"Data has been saved to {filename}")
```

פונקציה זו מקבלת את הנתונים שאספה הפונקציה הקודמת ושומרת אותם בקובץ Excel בשם שבחרנו בשורה 72 (web_crawl_data.xlsx) אנו משתמשים בספריית pandas כדי ליצור DataFrame מהנתונים שקיבלנו ובנוסף גם לשמירת הקובץ כקובץ Excel.

.6

```
79  def main():
80      """The main entry point of the script."""
81      print('Starting web crawling...')
82      base_url = "https://www.yahoo.com/"  # put your address here
83
84      extracted_data = crawl_website(base_url)
85      save_data_to_excel(extracted_data)
```

פונקציה זו היא פונקיית הכניסה הראשית של הסקריפט, אנו מגדירים את כתובת ה-URL שנרצה ממנה לשלוף מידע (אני בחרתי באתר www.yahoo.com) ולאחר מכן קוראים לפונקציה crawl_website כדי לאסוף נתונים מהאתר ולבסוף נקרא לפונקציה save_data_to_excel כדי לשמור את הנתונים שנאספו לקובץ excel.

.7

```
88  if __name__ == "__main__":
89      main()
90
```

קטע קוד זה נועד כדי להבטיח שפונקציית ה-main תופעל רק כאשר הסקריפט רץ ישירות ולא כשמיובא כמודל.