

تصميم تطبيق ويب كدسة كاملة باستخدام الخادم Flask في بايثون

ملخص

في العصر الراهن أصبحت تطبيقات الويب في مختلف مجالاتها حاجة أساسية لا يمكن الاستغناء عنها، وقد ترتب على هذا الشيء زيادة في الطلب ضمن سوق العمل على المبرمجين الخبراء في هذا المجال. يتم تقسيم تصميم مواقع الويب إلى قسمين أساسيين هما الواجهة الأمامية والواجهة الخلفية، والتعامل معهما في وقت واحد يسمى الكدسة الكاملة. في هذه المقالة سنقوم بتصميم تطبيق كدسة كاملة باستخدام لغة بايثون والمكتبات اللازمة لهذا المشروع وسنقوم بتصميم الواجهة الأمامية بشكل مدونة شخصية بسيطة نضع فيها بعض المعلومات الأساسية عن الباحثين.

Full Stack Web Application Using Python Flask Server

Abstract: In the current era, web applications in its various fields have become a basic and indispensable need, and this has resulted in an increase in the demand within the labor market for programmers who are experts in this field. Website design is divided into two main parts, the front end and the back end, and dealing with them simultaneously is called the full stack. In this article we will design a full stack application using Python language and the libraries needed for this project and design the front end in the form of a simple personal blog in which we will put some basic information about the researchers.

مقدمة:

تعريف : الموقع الإلكتروني بالإنجليزية (WEB Site) هو عبارة عن موقع مركزي يضم عدداً من صفحات الويب المرتبطة ببعضها البعض، و التي عادة ما يمكن الوصول إليها من خلال الصفحة الرئيسية (Home page) ، أما صفحة الويب فهي عبارة عن ملف مكتوب بلغة ترميز النص الفائق (HTML) ، ويمكن أن يتضمن نصوصاً و صوراً بالإضافة إلى وصلات لصفحات أخرى. يمكن الوصول إلى صفحة الويب بواسطة متصفح الويب (Web Browser) عن طريق كتابة عنوانها فيه. تم بناء أول موقع إلكتروني عام 1991 في مختبر سيرن (CERN) على يد مخترع لغة الترميز النص الفائق تيم بيرنرز لي (Tim Berners-Lee) ويمكن الوصول إلى هذا الموقع حالياً من خلال الرابط (<http://info.cern.ch>) بشكل عام. يتم إنشاء المواقع الإلكترونية الموجودة على شبكة الإنترنت عادة من قبل الشركات أو الحكومات أو حتى الأفراد. اخترع العالم البريطاني تيم بيرنرز لي شبكة الويب العالمية (WWW) في عام 1989 أثناء عمله في CERN. تم تصميم الويب وتطويره في الأصل لتلبية الطلب على تبادل المعلومات الآلي بين العلماء في الجامعات والمعاهد حول العالم.

هناك العديد من المواقع الإلكترونية على الإنترنت ويمكن تقسيمها إلى أحد أنواع الفئات التالية، فمن الممكن أن يكون الموقع مُحرك بحث أو موقع بريد إلكتروني أو مدونة أو منتدى، ومن الممكن أن يقع الواحد تحت فئات متعددة. فيما يلي سوف نذكر لكم أهم الفئات والأنواع من المواقع الإلكترونية:

*المدونة Blog : أي موقع على شبكة الإنترنت، ويتم إنشاؤه عادة من قبل فرد واحد بهدف الحفاظ على أشياء تهمّه، أو بهدف عرض مقالات، ومن الممكن إنشاؤه كذلك من قبل مؤسسة بهدف المشاركة في الأفكار والمعلومات حول المجالات التي تتخصص بها هذه المؤسسات .

*موقع للأعمال والشركات التجارية: يتم إنشاء موقع ويب تجاري أو موقع ويب للشركة؛ لتوفير معلومات عن الشركة أو المؤسسة، والوصول إلى العملاء والشركات والمعملاء المحتملين.

*مواقع التواصل الاجتماعي: يكون الهدف الأساسي من مواقع التواصل الاجتماعي هو دخول الزائرين والتفاعل داخل الموقع عن طريق الدردشة، وكتابة ما يخطر في بالهم، وتلقي الإعجاب والتعليقات على ما يكتبون، وغالباً ما تكون معظم خدماته مجانية، وأشهر مثال لمواقع التواصل الاجتماعي هو فيس بوك Facebook.

*مواقع محركات البحث Search Engine : تكون مواقع محركات البحث هي الوجهة الرئيسة لأي شخص يريد الوصول لشيء معين على الإنترنت، مثل الوصول إلى موقع معين أو خدمة معينة، أو السؤال عن معلومات مختلفة بهدف معرفة معلومات تخص العديد من المواضيع، ويعد من أشهر مواقع محركات البحث موقع جوجل Google.

*مواقع التجارة الإلكترونية E-Marketing : هي مواقع تم إنشاؤها بهدف بيع سلعة أو خدمة معينة عبر الإنترنت، بحيث يعد كل من amazon.com و alibaba.com من أشهر هذه المواقع على مستوى العالم المختصة بالتجارة الإلكترونية .

*مواقع المزادات: يختلف موقع المزادات شكلاً وتقسيمياً عن موقع التجارة الإلكترونية، حيث يسمح موقع المزادات للمستخدمين عرض السلع الخاصة بهم على الموقع، وعمل مزاد علني على السلعة في مقابل أن يأخذ هذا الموقع عمولة من نسبة البيع، ويعد موقع eBay.com أحد أشهر مواقع المزادات العلنية على الإنترنت .

*المواقع الحكومية: هي مواقعُ تنشأ من قبل الحكومات لتقديم الخدمات للمواطنين، وتعريف المواطنين بكل ما هو جديد داخل الدولة، وتستخدم للترويج السياحي للمعالم السياحية الموجودة داخل الدولة؛ لتعريف السائحين بها .

*مواقع الألعاب: هو موقع ويب يتم إنشاؤه خصيصاً لألعاب الفيديو Video Games ، ويمكن للمستخدمين ممارسة ألعابهم المفضلة على شبكة الويب Online ، دون الحاجة لتنزيل اللعبة على هواتفهم أو أجهزة الكمبيوتر، وغالباً ما يتم بناء هذه الألعاب عبر الإنترنت باستخدام لغات مثل HTML أو Flash أو Java.

* مواقع مشاركة الوسائط: هو موقع مُخصَّص لوضع الوسائط المرئية والمسموعة والصور عليه، ومشاركتها من قبل المستخدمين، ويعدُّ يوتيوب YouTube أشهر موقع لمشاركة الوسائط المرئية، وساوند كلاود Sound Cloud أشهر موقع لمشاركة الوسائط المسموعة، وموقع فليكر Flickr الأشهر لمشاركة الصور .

*مواقع البريد الإلكتروني: هو موقع إلكتروني على الويب تمكّن المستخدم من إرسال واستقبال رسائل البريد الإلكتروني عليه من دون الحاجة إلى برمجيات، وتعتبر مواقع البريد الإلكتروني هي الأكثر استخداماً والأشهر على الإطلاق من قبل كل من المستخدمين العاديين والشركات المتوسطة، ومن أشهر هذه المواقع Gmail وHotmail وYahoo.

* مواقع الأخبار: هو موقع إلكتروني يتم إطلاق محتواه من قبل فريق التحرير الخاص بالموقع، ويمكن للمستخدمين متابعة آخر الأخبار، والأخبار العاجلة بمجرد نشرها من قبل فريق التحرير، وتوجد مواقع إخبارية متخصصة في نوع واحد من الأخبار مثل الأخبار الاقتصادية، وهناك مواقع إخبارية تقوم بنشر أخبار متنوعة، مثل الأخبار الرياضية والسياسية والمالية والاقتصادية وغيرها

*المواقع الخبيثة: هو أي موقع إلكتروني أنشئ بغرض إصابة جهاز كمبيوتر آخر أو هاتف، أو جمع البيانات الشخصية الخاصة بالمستخدمين، أو تم إنشاؤه بغرض نشر برامج ضارة، أو برامج تجسس، أو فيروسات مثل فيروس حصان طروادة Trojan ، وتقوم هذه المواقع بتنزيل برامج ضارة على الجهاز المستهدف، وتقوم بسرقة ملفات تعريف المستخدم التي يستخدمها للدخول على الإنترنت. هناك أيضاً مواقع خبيثة تسمى مواقع التصيد، ويتم تصميم هذه المواقع على هيئة مواقع رسمية، على سبيل المثال يتم إنشاء موقع بنفس شكل موقع البنك الذي يتعامل معه المستخدم، وذلك من أجل سرقة بياناته المالية. وهناك نوع آخر من المواقع الخبيثة وهي مواقع الأخبار المزيفة، ويتم إنشاؤها بغرض بث الأخبار غير الحقيقة، والمساعدة في بثّ الخوف والأكاذيب بين المواطنين.

*المواقع الدراسية: يتم إنشاء هذه المواقع لفئة معينة فقط، مثل موقع ويب خاص بجامعة أو موقع ويب خاص بمدرسة، ويتم في هذا الموقع نشر محتوى يهتم فقط الطلاب المنتسبين لتلك الجامعة أو المدرسة.

* مواقع مساعدة: هو موقع يتم إنشاؤه خصيصاً لكي يذهب إليه المستخدم وي طرح أسئلته، ويقوم شخص آخر أو مجموعة من الأشخاص بالرد على تلك الأسئلة، وتقوم تلك المواقع على فكرة التشاركية حيث يستطيع أي شخص المشاركة في طرح الأسئلة، أو اجابتهـا.

أهمية البحث وأهدافه:

سنقوم بهذا البحث بإنشاء مدونة شخصية بسيطة وسنعمد على أساسيات تصميم المواقع ولغاتهما مثل html و css بالنسبة للفرونت ايند والـ Flask بالنسبة للباك ايند. وتمكن أهمية هذا البحث في فهم أساسيات تصميم المواقع ومعرفة كيف نبدأ في سوق العمل.

أدوات البحث:

- HTML (Hyper Text Markup Language) لغة النص التشعبي.
- (Cascading Style Sheets) css
- Flask server
- Python

منهجيات البحث:

1- HTML

هي لغة ترميز تستخدم في إنشاء وتصميم صفحات ومواقع الويب، وتعتبر هذه اللغة من أقدم اللغات وأوسعها استخداماً في تصميم صفحات الويب HTML. هيكل صفحة الويب وتعطي متصفح الإنترنت وصفاً لكيفية عرضه لمحتوياتها، يمكن أن تساعد تقنيات مثل (CSS) ولغات البرمجة النصية مثل جافا سكريبت تستقبل متصفحات الويب مستندات HTML من خادم الويب أو من نظام الملفات وتعرضها، ووظيفة لغة HTML هي وصف بنية صفحات الويب هيكلياً. العناصر في HTML هي البنية الأساسية لبناء مستندات HTML ، إذ نستطيع عبرها إضافة الصور والكائنات التفاعلية مثل النماذج أو ملفات الفيديو والصوت؛ وتستطيع أيضاً إنشاء مستندات منظمة عبر استخدام وسوم للتصريح عن الفقرات والعناوين والروابط والاقتباسات والجداول وغيرها. يمكن للغة HTML أن تُضمّن برامج مكتوبة بلغات مثل جافا سكريبت لتعديل سلوك ومحتوى صفحات الويب؛ وإضافة شفرات CSS تؤدي إلى تعريف شكل وتخطيط المحتوى.

صفحة الـ html لها الهيكلية الأساسية التالية:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
```

```
</head>
<body>
  <h1>This is a heading</h1>
  <p>Hello World!</p>
</body>
</html>
```

- تطبيقات HTML

كما ذكرنا سابقاً ، تعد HTML واحدة من أكثر اللغات استخداماً على الويب. سأقوم بإدراج عدد قليل منهم هنا:

- تطوير صفحات الويب - يتم استخدام HTML لإنشاء الصفحات التي يتم تقديمها عبر الويب. تحتوي كل صفحة ويب تقريباً على علامات html لعرض تفاصيلها في المتصفح.
- التنقل عبر الإنترنت - يوفر HTML علامات تُستخدم للتنقل من صفحة إلى أخرى وتستخدم بكثرة في التنقل عبر الإنترنت.
- واجهة مستخدم مستجيبة - تعمل صفحات HTML في الوقت الحالي بشكل جيد على جميع الأنظمة الأساسية أو الأجهزة المحمولة أو علامات التبويب أو سطح المكتب أو أجهزة الكمبيوتر المحمولة بسبب استراتيجية التصميم سريعة الاستجابة.
- يمكن إتاحة صفحات HTML للدعم غير المتصل بالإنترنت بمجرد تحميلها في وضع عدم الاتصال على الجهاز دون الحاجة إلى الاتصال بالإنترنت.
- تطوير الألعاب - تتمتع HTML5 بدعم أصلي لتجربة غنية وهي الآن مفيدة في مجال تطوير الألعاب أيضاً.

-2 CSS:

هي لغة تنسيق لصفحات الويب تهتم بشكل وتصميم المواقع، صممت خصيصاً لفصل التنسيق(الألوان-الخطوط-الأزرار - ...)، بلغة مثلاً HTML وينطبق ذلك على الألوان والخطوط والصور والخلفيات التي تستخدم في الصفحات، بمرونة وسهولة تامة. هذه التقنية تعنى بالمظهر الكلي لصفحات مواقع الويب من ألوان وصور وغيره. ويمكن إضافته للصفحة بعدة طرق أفضلها التضمين الخارجي بكتابة شفرة CSS ملف منفصل. وقد تم تطويرها لتصل حالياً إلى CSS3 والذي أضيف إليه عدة إضافات لم تكن متاحة في الإصدار السابق وكان إنجازها يتم فقط JQuery.

:Flask server -3

Flask هو إطار عمل ويب صغير مكتوب بلغة Python. تم تصنيفها على أنها إطار مصغر لأنها لا تتطلب أدوات أو مكتبات معينة. لا يحتوي على طبقة تجريد لقاعدة البيانات أو التحقق من صحة النموذج أو أي مكونات أخرى حيث توفر مكتبات الجهات الخارجية الموجودة مسبقاً وظائف مشتركة. ومع ذلك، يدعم Flask الملحقات التي يمكنها إضافة ميزات التطبيق كما لو تم تنفيذها في Flask نفسه. توجد امتدادات لوضع هيكلية الكائنات، والتحقق من صحة النموذج، ومعالجة التحميل، والعديد من تقنيات المصادقة المفتوحة والعديد من الأدوات ذات الصلة بإطار العمل المشترك.

تشمل التطبيقات التي تستخدم إطار العمل هذا LinkedIn و Pinterest.

:Full Stack -4

تشير تقنية Full stack إلى العمق الكامل لتطبيق نظام الكمبيوتر، ويعمل مطورو Full stack على مجالين منفصلين لتطوير الويب: الواجهة الأمامية والنهاية الخلفية.

تتضمن الواجهة الأمامية كل شيء يمكن للعميل أو عارض الموقع رؤيته والتفاعل معه. على النقيض من ذلك، تشير النهاية الخلفية إلى جميع الخوادم وقواعد البيانات والبنية الداخلية الأخرى التي تقود التطبيق؛ عادةً ، لا يتفاعل المستخدم النهائي أبداً مع هذا المجال بشكل مباشر.

أسهل طريقة لوضع Full stack في منظورها الصحيح هي تخيل مطعم. تشمل الواجهة الأمامية مناطق جلوس مريحة ومُزينة جيداً حيث يستمتع الزوار بطعامهم. يشكل المطبخ والمخزن "النهاية الخلفية" وعادة ما يتم إخفاؤها بعيداً عن رؤية العميل. يقوم الطهاة (المطورو) بجمع المواد المخزنة بشكل دائم من المخزن (قاعدة البيانات) وإجراء العمليات عليها في المطبخ (الخادم)، ثم يقدمون وجبات جاهزة (معلومات) للمستخدم.

يعمل مطورو الواجهة الأمامية على تحسين الأجزاء المرئية من التطبيق لمتصفحات الويب والأجهزة المحمولة. عادةً ما يتم إنشاء منصات الواجهة الأمامية باستخدام HTML و CSS و JavaScript ؛ ومع ذلك ، يمكن أيضاً إجراؤها عبر مكتبات التعليمات البرمجية المعبأة مسبقاً أو أنظمة إدارة المحتوى مثل WordPress. على النقيض من ذلك، يقوم مطورو الواجهة الخلفية بتحسين رمز البرنامج الذي يتصل بالخوادم أو قواعد البيانات أو غيرها من البرامج الاحتكارية التي تنقل المعلومات إلى واجهات الواجهة الأمامية.

يُطلق على هؤلاء المطلعين على كل من الواجهة الأمامية والخلفية مطوري Full stack، مما يعني أنهم على دراية جيدة في كلا المجالين.

نشأ مصطلح "مطور Full stack" خلال الأيام الأولى للويب، عندما كانت مواقع الويب صغيرة وغير معقدة بما يكفي للسماح لشخص واحد بمعالجة كل جانب من جوانب بناء الموقع. ولكن في العقود التي انقضت منذ تلك الأيام الأولى، أصبحت شبكة

الويب أكثر تعقيداً من أي وقت مضى. أدى ظهور التعلم الآلي والحوسبة التنبؤية والتصميم سريع الاستجابة إلى جعلها صعبة - ولكنها ليست مستحيلة! - لمطور واحد للتعامل مع كل جانب من جوانب بناء وتصميم موقع أو تطبيق.

اليوم ، غالباً ما تعتمد الشركات الحديثة على فرق كاملة من المطورين لتشغيل معدات الشبكات ، والعمل مع الأجهزة الافتراضية، وإدارة قواعد البيانات الضخمة. يستغرق الأمر وقتاً لتطوير فهم شامل لجميع هذه التقنيات الناشئة. لهذا السبب، فإن المطورين الذين يقومون بذلك متعددون بما يكفي للانتقال بسلاسة بين تطوير الواجهة الأمامية والخلفية والاضطلاع بأي مهمة قد يحتاج فريقهم إلى معالجتها.

وفقاً لاستطلاع أجرته Stack Overflow 2020 على 65000 مطور في جميع أنحاء العالم، ما يقرب من 54.9 بالمائة، يتم تحديدهم على أنهم Full stack.

(على النقيض من ذلك ، عرّف 55.2 بالمائة من الذين شملهم الاستطلاع أنفسهم على أنهم مطورو الواجهة الخلفية ، و 37.1 بالمائة عرّفوا أنفسهم كمطورين للواجهة الأمامية ، وزعم 19.2 بالمائة فقط أنهم مطورو هواتف محمولة.)

القسم العملي:

نبدأ بالواجهة الخلفية:

نقوم بإنشاء السيرفر Flask باستخدام الـ pycharm وفق الآتي:

```
from flask import Flask, render_template, request

app = Flask(__name__)

database={"yara": "12345", "zubaida": "2222"}
```

نستورد الكلاس Flask من أجل إنشاء السيرفر والتوابع render_template من أجل جعل الاستجابة هي صفحة html والتابع request من أجل التعامل مع الطلب القادم من الواجهة الأمامية.

بعدها ننشئ غرض app من الكلاس Flask. ونعرف قاعدة بيانات على شكل متغير من نوع dictionary تحتوي اسم المستخدم وكلمة المرور لكل مستخدم.

```
if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=3000)
```

التعليمة السابقة من أجل تشغيل السيرفر في وضع التصحيح من أجل تتبع الخطأ مع ضمان عدم توقف البرنامج.


```
@app.route("/")
def home():
    return render_template("login.html")
```

التابع home يتم تنفيذه عندما يتم طلب الدومين الرئيسي ويتم التقاط الطلب عن طريق المنسق app.route ويقوم هذا التابع بإرجاع قيمة إلى المستخدم وهي عبارة عن صفحة تسجيل الدخول إلى الموقع وذلك عن طريق التابع render_tempplate.

```
@app.route("/login", methods=["POST"])
def login():
    name=request.form['username']
    password=request.form['password']
    if name in database:
        if database[name]==password:
            return render_template("homepage.html")
        else:
            return render_template("login.html", info="wrong
password")
    else:
        return render_template("login.html", info="wrong
password or username")
```

التابع login يتم تنفيذه عند ورود طلب من url عبارة عن الدومين الأساسي ملحق بالدومين الفرعي login ويكون الطلب من نوع post أي يحوي قيم ويختبر هذا التابع إن كان اسم المستخدم وكلمة المرور موجودين ضمن قاعدة البيانات ليعيد الصفحة homepage أو هنالك خطأ في اسم المستخدم أو كلمة المرور فيعيد نفس صفحة تسجيل الدخول مع متغير اسمه info يحوي رسالة يتم عرضها على صفحة الـ .html.

```
@app.route('/signup', methods=["POST", "GET"])
def signup():
    if not request.form:
        return render_template("signup.html")
    database[request.form['username']] = request.form['password']
    return render_template("login.html")
```

التابع singup من أجل إنشاء حساب على الموقع أي إضافة اسم المستخدم وكلمة المرور إلى قاعدة البيانات ويعيد صفحة تسجيل الدخول أيضاً.

```
@app.route("/home")
def homepage():
    return render_template("homepage.html")
@app.route("/about")
def about():
    return render_template("about.html")
```

التابعين homepage و about يعيدان الصفحة home و الصفحة about بالترتيب.

بالنسبة للواجهة الأمامية:

بالنسبة لصفحة تسجيل الدخول نقوم بعرض جزء من الكود المصدري للصفحة وهو الجزء الخاص باسم المستخدم وكلمة المرور:

```
<form action="/login" method="POST" class="form login">

  <div class="form__field">
    <label for="login_username">
      <svg class="icon">
        <use xlink:href="/icon.gif"></use>
      </svg><span class="hidden">Username</span></label>
      <input autocomplete="username" id="login_username" type="text"
name="username" class="form__input" placeholder="Username" required>
    </div>

    <div class="form__field">
      <label for="login_password"><svg class="icon">
        <use xlink:href="#icon-lock"></use>
      </svg><span class="hidden">Password</span></label>
      <input id="login_password" type="password" name="password"
class="form__input" placeholder="Password" required>
    </div>

    <div class="form__field">
      <input type="submit" value="Sign In">
    </div>

  </form>
```

الوسم form خاص بتشكيل واجهة إدخال بيانات من أجل تعبئتها وإرسالها إلى المخدم حيث يقوم بطلب الصفحة **"/login"** ونوع الطلب هو **"POST"**

بالنسبة لخانة اسم المستخدم نطبق تنسيق معين أي شكل الواجهة هو كود الـ CSS الموافق لـ **form__field**

نعرض كود الـ CSS :

```
form__field {
  display: flex;
}
```

والذي يعني أن نمط العرض هو flex لهذا الجزء من صفحة الويب. وضعنا كود من أجل الزر الذي سيتم النقر عليه لإرسال الفورم وذلك من خلال الوسم input والنوع **"submit"**

يتم وضع اسم المستخدم في متغير اسمه username وذلك من خلال الصيغة التالية: **name="username"**

يتم وضع كلمة السر في متغير اسمه password وذلك من خلال الصيغة التالية: **name="password"**

ويتم الوصول إلى هذين المتغيريين في السيرفر باستخدام المكتبة request.

```

<body>
  <header>
    <div class="wrapper">
      <h1>Programming Network<span class="color">.</span></h1>
      <nav>
        <ul>
          <li><a href="/home">Home</a></li>
          <li><a href="/about">About</a></li>
        </ul>
      </nav>
    </div>
  </header>

  <<div id="hero-image">
    <div class="wrapper">
      <h2><strong>Programming Network Project</strong><br/>
      الشبكات برمجة مشروع
    </div>
  </div>

  <div id="features">
    <div class="wrapper">
      <ul>
        <li class="feature-1">
          <h4>علوش صدوح يارا</h4>
          <p>1824 الجامعي الرقم</p>
        </li>
        <li class="feature-2">
          <h4>سلمان محمد زبيدة</h4>
          <p>1465 الجامعي الرقم</p>
        </li>
        <li class="feature-3">
          <h4>الدكتور إشراف</h4>
          <p>عيسى مهند</p>
        </li>
      <div class="clear"></div>
    </ul>
  </div>
</div>

  <div id="primary-content">
    <div class="wrapper">
      <article>
        <h3>الصفحة هذه محتويات</h3>
        <p>بهم الخاصة الجامعية والأرقام الطلاب أسماء على الصفحة هذه تحوي</p>
        <p>المشروع هذا اعجابكم ينال أن ونتمنى أخرى لصفحات روابط على وأيضاً</p>
        <a href="#"></a>
      </article>
    </div>
  </div>

  <div id="cta">

```

```

<div class="wrapper">
  <h3></h3>المتواضع موقعنا في بكم وسهلاً أهلاً</h3>
  <p>HTML: Hyper Text Markup Language، وتعتبر الويب، ومواقع الويب صفحات تصميم و إنشاء في تستخدم رقم لغة هي، HTML تعطي. الويب صفحات تصميم في استخداماً وأوسعها اللغات أقدم من اللغة هذه تعلمه فهي للمحتوياتها، عرضه لكيفية وصفا الإنترنت متصفح وتعطي الويب صفحة هيكل ما يعرف HTML الـ وتستخدم. الكثير ذلك وغير فقرة وتلك رئيسي عنوان هذا بأن أكبر علامتي بين توضع الوسوم هذه المتصفح، إلى التعليمات لإصدار ('tags') بالوسوم
  <a href="#" class="button-2">بنا هيا</a>
</div>
</div>
<footer>
  <div class="wrapper">
    <div id="footer-info">
      <p>Copyright 2022 Communications and electronics engineering. All rights reserved.</p>
      <p><a href="#">I</a> <a href="#">الخصوصية</a> <a href="#">شروط</a> </p>
    </div>
    <div id="footer-links">
      <ul>
        <li><h5>المؤسسة</h5></li>
        <li><a href="#">والالكترونيات الاتصالات هندسة</a></li>
        <li><a href="#">الشبكات برمجة</a></li>
        <li><a href="#">عيسى مهند د.</a></li>
      </ul>
      <ul>
        <li><h5>الطلاب أسماء</h5></li>
        <li><a href="#">سلمان محمد زبيدة</a></li>
        <li><a href="#">علوش صدوح يارا</a></li>
      </ul>
    </div>
    <div class="clear"></div>
  </div>
</footer>
</body>
</html>

```

نقوم بتقسيم الصفحة إلى أجزاء ضمن الوسم body من أجل الترتيب ونضع تنسيق CSS لكل جزء على حدى:

```

<style>
  body {
    background-color: #fff;
    color: #777;
    font: normal 15px "Helvetica Neue", Arial, Helvetica, Geneva, sans-serif;
  }
  p {
    line-height: 20px;
    margin-bottom: 20px;
  }

```

```

h1 {
  font-family: 'Crete Round', serif;
  font-weight: bold;
  color: #444;
  font-size: 45px;
  margin-bottom: 20px;
}
h2 {
  font-weight: 300;
  color: #444;
  font-size: 55px;
  text-transform: uppercase;
  text-align: center;
  margin-bottom: 20px;
}
h3 {
  font-size: 30px;
  color: #444;
  font-weight: bold;
  text-transform: uppercase;
  text-align: center;
  margin-bottom: 20px;
}
h4 {
  font-size: 24px;
  color: #444;
  font-weight: bold;
  text-transform: uppercase;
  margin-bottom: 20px;
}
h5 {
  font-size: 15px;
  color: #444;
  font-weight: bold;
  text-transform: uppercase;
}
a {
  text-decoration: none;
  color: #444;
}
a:hover {
  color: #02b8dd;
}
strong {
  font-weight: bold;
}
small {
  font-size: 13px;
  color: #777;
  font-style: italic;
}
.clear {
  clear: both;
}
.wrapper {
  margin: 0 auto;
  padding: 0 10px;
}

```

```

        width: 940px;
    }
    header {
        height: 120px;
    }
    header h1 {
        float: left;
        margin-top: 32px;
    }
    header h1 .color {
        color: #02b8dd;
    }
    header nav {
        float: right;
    }
    header nav ul li {
        float: left;
        display: inline-block;
        margin-top: 50px;
    }
    header nav ul li a {
        color: #444;
        text-transform: uppercase;
        font-weight: bold;
        display: block;
        margin-right: 20px;
    }
    #hero-image {
        height: 580px;
        padding-top: 1px;
        background: #e8eced url('2.jpg') no-repeat center;
    }
    #hero-image h2 {
        margin: 180px 0 40px 0;
    }

    #features ul {
        margin: 80px 0;
    }
    #features ul li {
        width: 300px;
        padding-top: 140px;
        float: left;
        margin-right: 10px;
        text-align: center;
    }

    #primary-content {
        background-color: #f8fafa;
        padding: 60px 0;
        text-align: center;
    }
    #primary-content h3 {
        display: block;
        margin: 0 auto 20px auto;
        width: 400px;
    }

```

```

        border-bottom: 1px solid #02b8dd;
        padding: 0 0 20px 0;
    }
    #primary-content a img {
        margin: 20px 0;
    }

    article h4 {
        border-bottom: 1px solid #02b8dd;
        padding-bottom: 20px;
    }
    .more-link {
        border: 1px solid #02b8dd;
        color: #02b8dd;
        padding: 6px 20px;
        border-radius: 3px;
    }
    .more-link:hover {
        background-color: #02b8dd;
        color: #fff;
    }
    #cta {
        padding: 60px 0;
        text-align: center;
    }
    #cta h3 {
        display: block;
        margin: 0 auto 20px auto;
        width: 400px;
        border-bottom: 1px solid #02b8dd;
        padding: 0 0 20px 0;
    }
    .button-2 {
        display: block;
        margin: 0 auto;
        border: 2px solid #02b8dd;
        color: #02b8dd;
        border-radius: 3px;
        width: 180px;
        height: 50px;
        font-size: 20px;
        line-height: 50px;
    }
    .button-2:hover {
        background-color: #02b8dd;
        color: #fff;
    }
    footer {
        padding: 60px 0;
        background-color: #f8fafa;
    }
    #footer-info {
        width: 380px;
        float: left;
        margin-top: 10px;
    }
    #footer-links {

```

```

        width: 520px;
        float: right;
    }
    #footer-links ul {
        width: 150px;
        float: left;
        margin-left: 20px;
    }
    #footer-links ul li {
        margin: 10px 0;
    }
</style>

```

باقي الصفحات تتبع الأسلوب ذاته.

النتائج والمناقشة:

نقوم بتشغيل برنامج السيرفر:

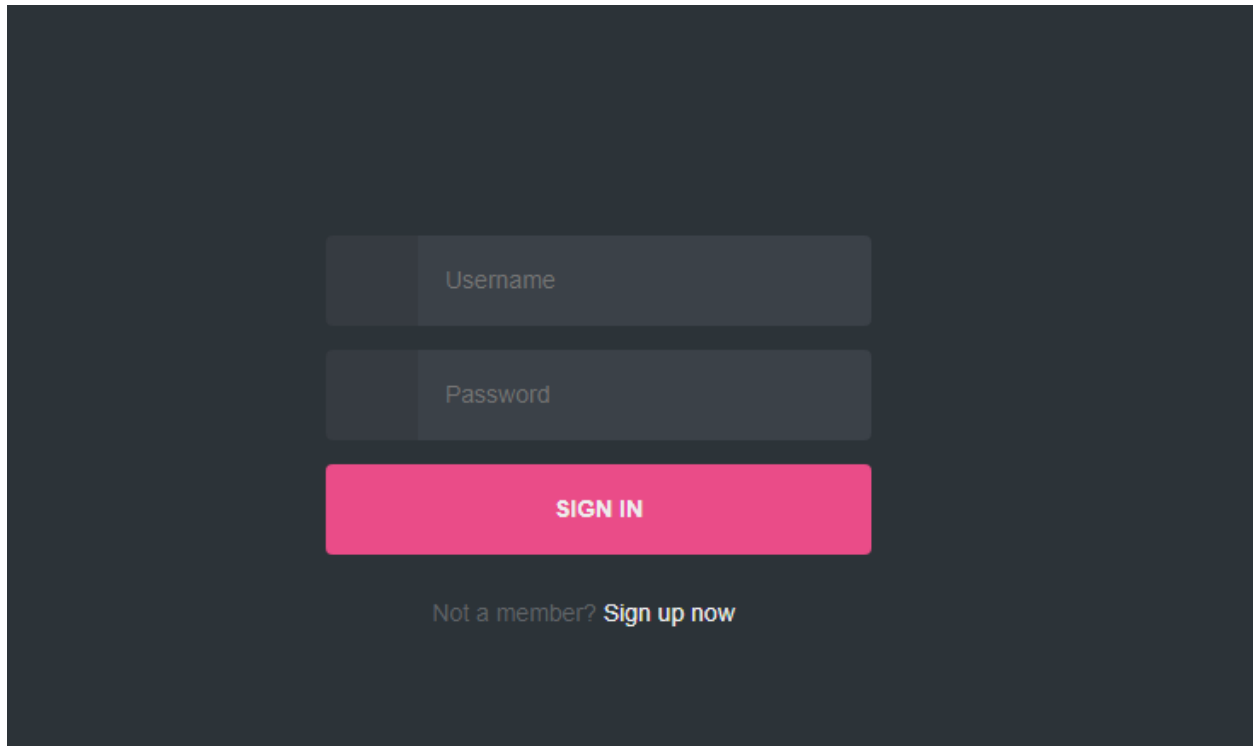
```

* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses (0.0.0.0)
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:3000
* Running on http://192.168.1.104:3000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 133-916-896

```

الشكل 1 تشغيل السيرفر

نقوم بالاتصال بالموقع على العنوان <http://127.0.0.1:3000>:



The image shows a dark-themed login interface. It features two input fields: 'Username' and 'Password', both with light gray placeholder text. Below these fields is a prominent pink button labeled 'SIGN IN' in white capital letters. At the bottom of the form, there is a link that reads 'Not a member? Sign up now'.

الشكل 2 نافذة تسجيل الدخول

يظهر على خرج السيرفر الطلب باستخدام التابع GET وكود الحالة 200 أي أن الطلب والاستجابة تمت بشكل صحيح:

```
* debug mode: on
* Running on all addresses (0.0.0.0)
  WARNING: This is a development server. Do not use it in a production deploym
* Running on http://127.0.0.1:3000
* Running on http://192.168.1.104:3000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 133-916-896
127.0.0.1 - - [28/May/2022 19:50:59] "GET / HTTP/1.1" 200 -
```

الشكل 3 خرج السيرفر بعد أول طلب

نقوم بتسجيل الدخول باستخدام اسم المستخدم وكلمة المرور الصحيحتين:

Programming Network.

HOME ABOUT



الشكل 4 الصفحة الرئيسية

نلاحظ في خرج السيرفر أنه تم إرسال الصورتان الموجودتان في المجلد static إلى الواجهة الأمامية:

```
-----
* Running on all addresses (0.0.0.0)
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:3000
* Running on http://192.168.1.104:3000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 133-916-896
127.0.0.1 - - [28/May/2022 20:06:57] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [28/May/2022 20:06:57] "GET /static/1.jpg HTTP/1.1" 304 -
127.0.0.1 - - [28/May/2022 20:06:57] "GET /static/2.jpg HTTP/1.1" 304 -
```

الشكل 5 خرج السيرفر

الاستنتاجات:

نستنتج أن لغة بايثون تقدم خدمة الويب بشكل جيد وذلك من خلال السيرفر flask وتتيح التعامل بسهولة مع الطلبات وبسرعة وبذلك نكون قد حققنا نظام Full stack يتكون من الواجهة الأمامية html و css والواجهة الخلفية flask server .

المراجع:

- 1- Murali, P., Linke, N. M., Martonosi, M., Abhari, A. J., Nguyen, N. H., & Alderete, C. H. (2019, June). Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)* (pp. 527-540). IEEE.
- 2- <https://kanchanardj.medium.com/how-to-add-images-to-html-in-a-flask-app-4dbcc92e3aeb>
- 3- <https://www.analyticsvidhya.com/blog/2022/01/develop-and-deploy-image-classifier-using-flask-part-2/>
- 4- <https://www.youtube.com/watch?v=w54WLGm4OrE>
- 5- <https://www.w3schools.com/css/>
- 6- <https://www.w3schools.com/html/>