# CS50x Iran

راهنمای پروژه Wiki

HARVARD
UNIVERSITY

# شروع کار

کد توزیع را از
https://cdn.cs50.net/web/2020/spring/projects/1/wiki.zip/
دانلود کنید.

# پیش زمینه

ویکی پدیا (Wikipedia) دانشنامه آنلاین رایگانی است که شامل دانشنامه های متنوع در زمینه های مختلف است .
هر دانشنامه را میتوان با بازدید از صفحه آن، مشاهده کرد. برای مثال، صفحه
https://en.wikipedia.org/wiki/HTML ویکی پدیا درباره HTM L اطلاعاتی به شما میدهد. توجه کنید که نام
صفحه درخواست شده ( ) HTML در مسیر wiki/HTML/ مشخص شده است.

همچنین متوجه هستید که محتوای صفحه فقط باید HTML باشد که مرورگرتان ارائه میکند.

# پیش زمینه

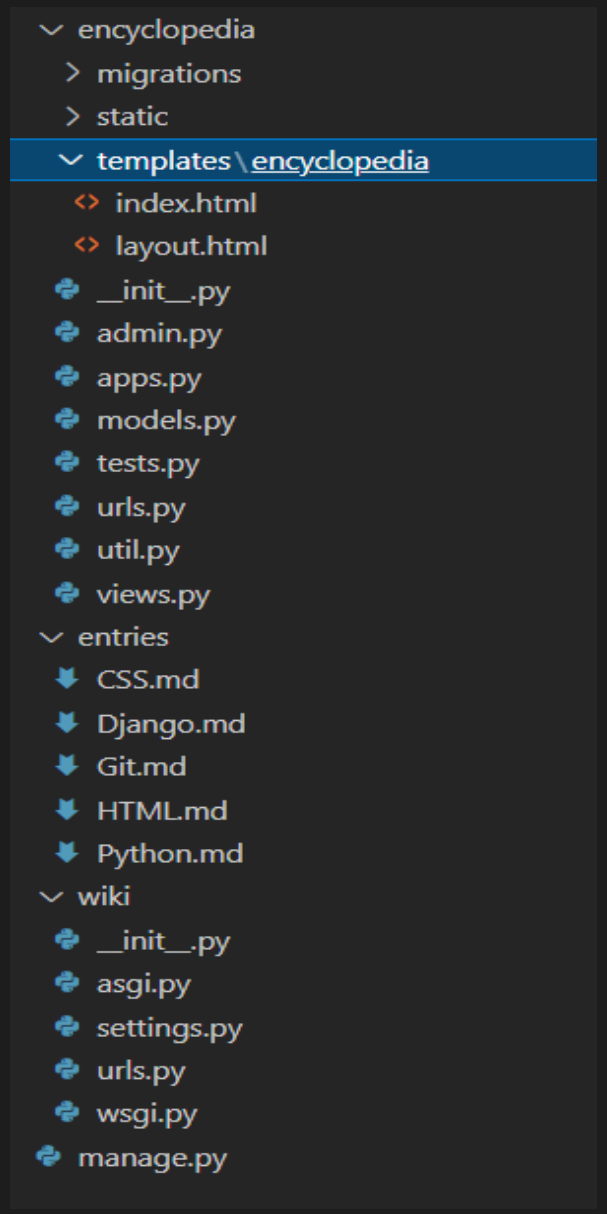
در عمل، اگر قرار باشد تمام صفحات ویکی پدیا با HTML نوشته شوند، کمکم خسته کننده میشود. در عوض، شاید خوب باشد که دانشنامه ها را با استفاده از زبان نشانه گذاری کاربر پسند و سبک تری ذخیره کرد.

ویکی پدیا از زبان نشانه گذاری Wikitext استفاده میکند، اما برای این پروژه، دانشنامه ها را با استفاده از زبان نشانه گذاری Markdown ذخیره خواهیم کرد .

# پیش زمینه

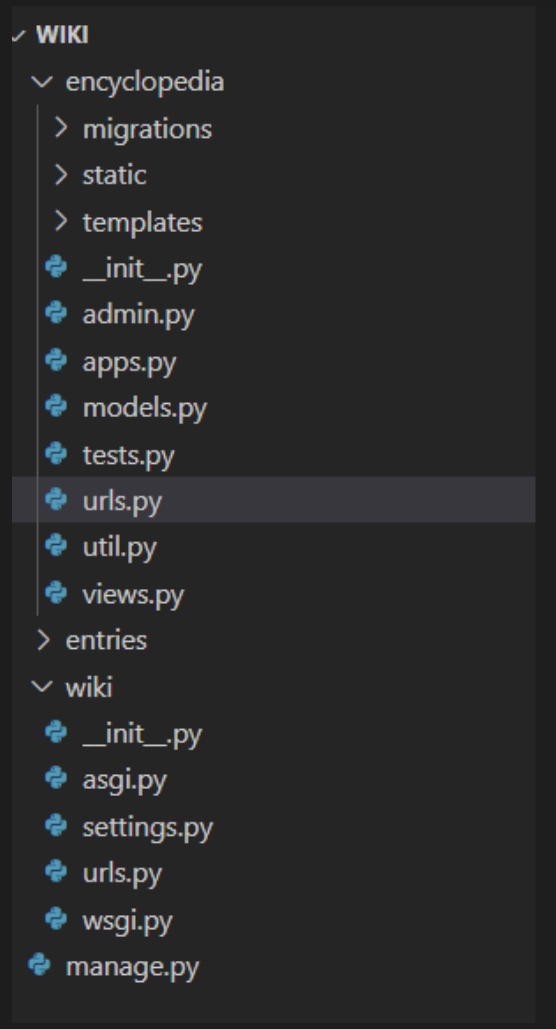
راهنمای Markdown گیتهاب را مطالعه کنید تا نحوه کار سینتکس Markdownرا درک کنید. به ویژه، توجه کنید که سینتکس Markdown در مورد عنوان ها، متن های توپر، لینک ها و فهرست ها به چه شکل است .

با داشتن یک فایل Markdown که تک تک دانشنامه ها را نشان میدهد ، میتوانیم نوشتن و ویرایش دانشنامه هایمان را کاربر پسندتر کنیم. با این حال، وقتی کاربر دانشنامه مان را مشاهده میکند، لازم است این Markdown را قبل از نمایش به کاربر به HTML تبدیل کنیم.




# درک کردن

در کد توزیع، پروژه جنگویی به نام wiki داریم که حاوی یک برنامه به نام encyclopedia است

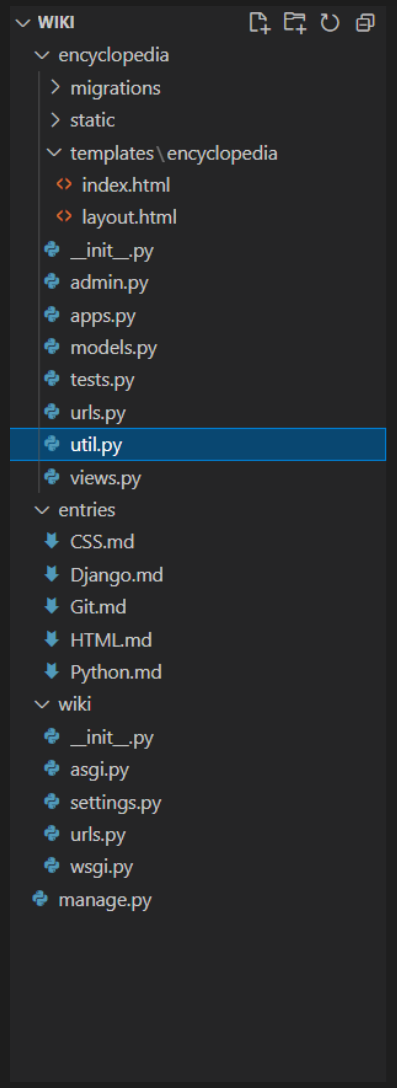



encyclopedia > urls.py > ...

```python
from django.urls import path

from . import views

urlpatterns = [
    path("", views.index, name="index")
]

```

## Urls.py

پیکربندی URL در این فایل تعریف شده است .
با یک مسیر پیشفرض شروع کرده ایم که با تابع views.index مرتبط است .




encyclopedia > util.py > ...

```python
import re

from django.core.files.base import ContentFile
from django.core.files.storage import default_storage


def list_entries():
    """
    Returns a list of all names of encyclopedia entries.
    """
    _, filenames = default_storage.listdir("entries")
    return list(sorted(re.sub(r"\.md$", "", filename)
                       for filename in filenames if filename.endswith(".md")))


def save_entry(title, content):
    """
    Saves an encyclopedia entry, given its title and Markdown
    content. If an existing entry with the same title already exists,
    it is replaced.
    """
    filename = f"entries/{title}.md"
    if default_storage.exists(filename):
        default_storage.delete(filename)
    default_storage.save(filename, ContentFile(content))


def get_entry(title):
    """
    Retrieves an encyclopedia entry by its title. If no such
    entry exists, the function returns None.
    """
    try:
        f = default_storage.open(f"entries/{title}.md")
        return f.read().decode("utf-8")
    except FileNotFoundError:
        return None
```

## encyclopedia/util.py

لازم نیست چیزی را در این فایل تغییر دهید ، سه تابع در این فایل وجود دارد که برای تعامل با دانشنامه ها مفید است.

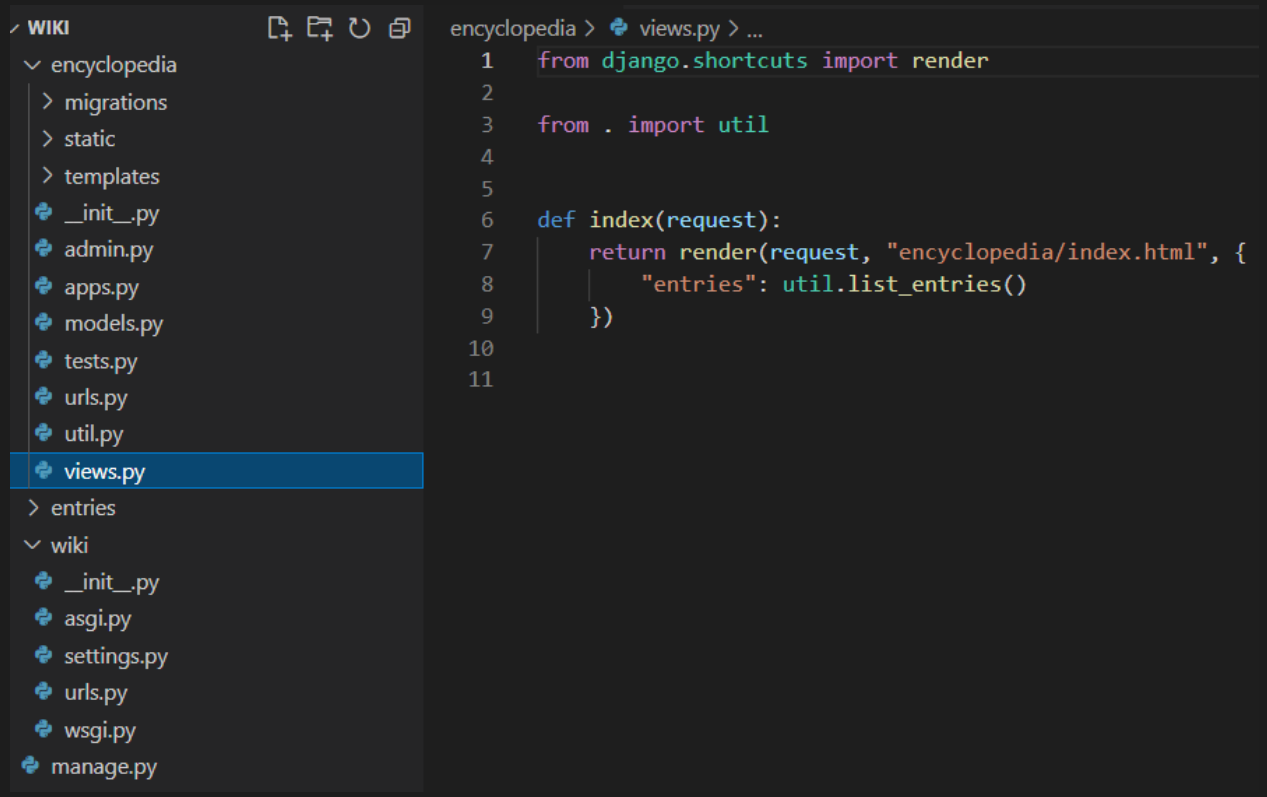
-تابع list_entries فهرست نام تمام دانشنامه هارا که قبلا ذخیره شده اند بر میگرداند.
_تابع save_entry دانشنامه جدید را با توجه به عنوان آن و محتوای Markdown ذخیره میکند.

## encyclopedia/util.py

ـتابع get_entry دانشنامه را براساس عنوان آن بازیابی می کند ، اگر دانشنامه ای وجود داشته باشد محتوای Markdown را برمیگرداند و اگر وجود نداشته باشد None برمی گرداند.
هر کدام از view هایی که مینویسید ممکن است از این توابع برای تعامل با دانشنامه ها استفاده کنند.

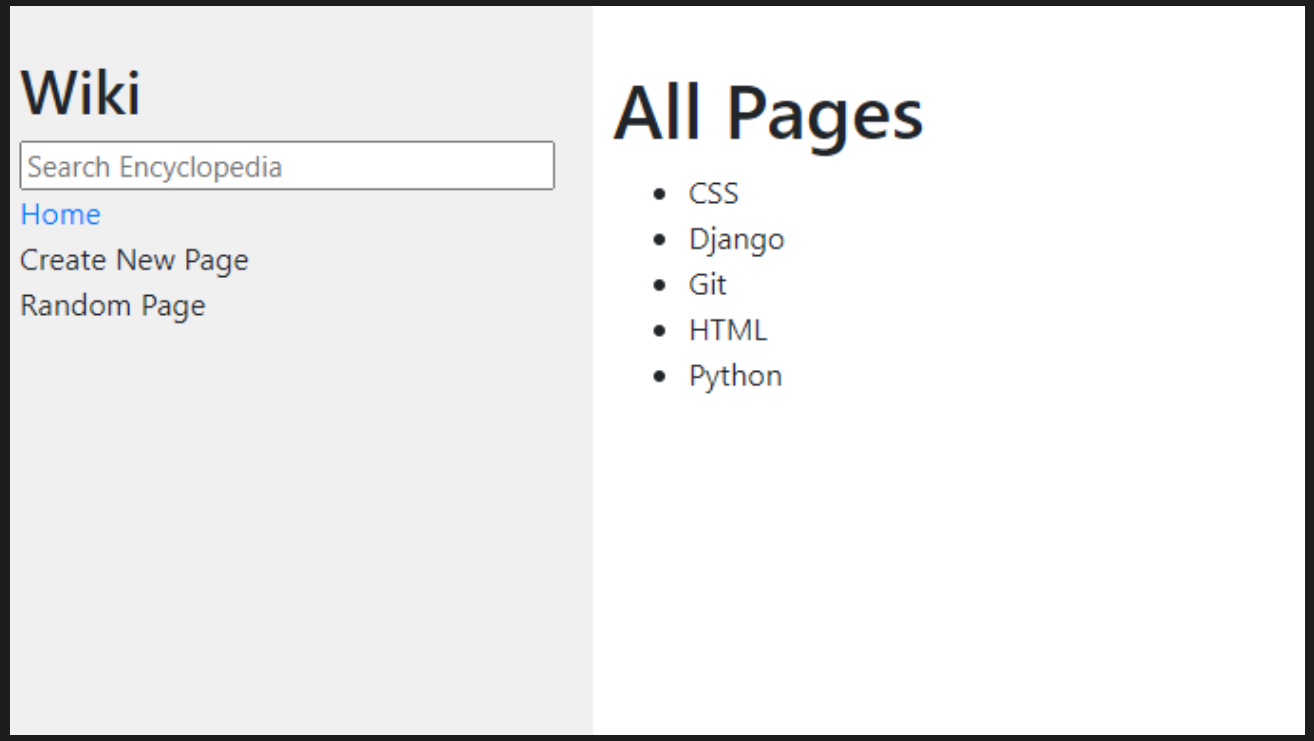
هر دانشنامه به صورت فایل Markdown در مسیر entries / ذخیره خواهد شد. اگر الان آن را بررسی کنید، چند دانش نامه ساده پیش ساخته مشاهده خواهید کرد. میتوانید چیزهای بیشتری اضافه کنید

## encyclopedia/views.py



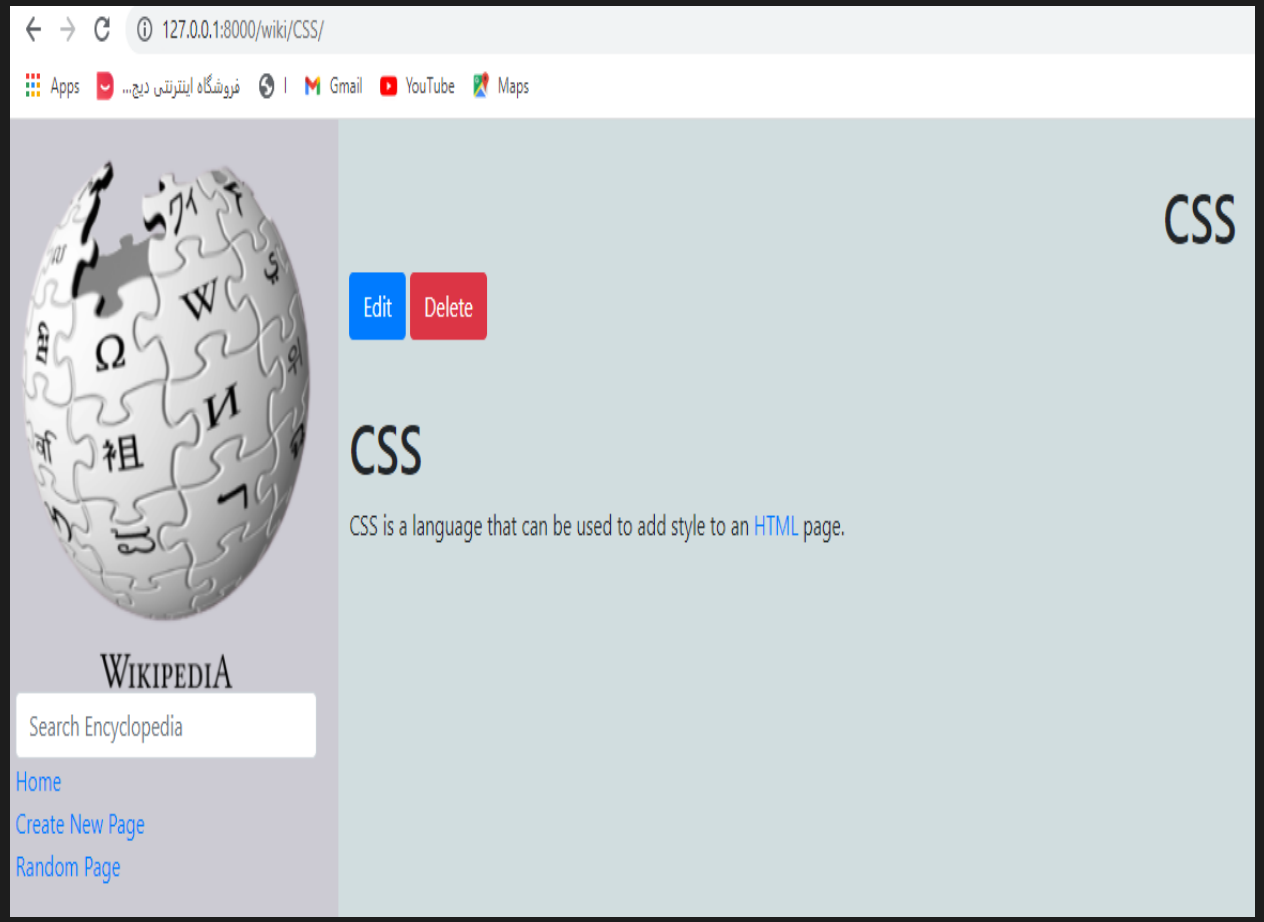

اینجا از تمام دانشنامه فقط یک View وجود دارد : index این view تمپلیت encyclopedia/index.html را برمیگراند و فهرستی از تملم دانشنامه ها را برای تمپلیت فراهم میکند. این فهرست با فراخوانی utile.list_entries بدست می آید.

## Templates/encyclopedia




میتوانید تمپلیت index.html را از encyclopedia/templates/encyclopedia/index.html بیابید. این تمپلیت از فایل پایه layout.html ارث میبرد و مشخص می کند که عنوان صفحه چگونه باید باشد ، در بدنه صفحه چه چیزی باید باشد : در این مورد فهرست نامرتبی از تمام دانشنامه ها را داریم. در عین حال ، layout.html ساختار کلی تر صفحه را تعریف میکند : هرصفحه یک ستون فرعی و فیلد جستجو (که فعلا کاری نمی کند)، لینک به صفحه اول ،و لینک های برای ساخت صفحه جدید یا بازدید از یک صفحه تصادفی دارد (این لینک ها هنوز کار نمی کنند).

# Entry Page




با بازدید از، wiki/TITLE که در آن TITLE برابر است با عنوان مدخل دانشنامه، باید
صفحهای نمایان شود که محتوای آن مدخل دانشنامه را نمایش میدهد.
این view باید محتوای دانشنامه را با فراخوانی تابع utilمناسب دریافت کند

اگر دانشنامه درخواستشده وجود نداشته باشد، باید صفحه خطایی به کاربر نمایش داده شود که بگوید

- صفحه درخواستشده پیدا نشده است

اگر این مدخل وجود داشته باشد، باید صفحهای به کاربر ارائه شود که محتوای آن دانشنامه را نمایش
دهد. عنوان صفحه باید شامل نام آن دانشنامه باشد.

# صفحه index


Index.html را بروزرسانی کنی تا بجای آنکه فقط نام تمام صفحات دانشنامه را فهرست کند ، کاربر بتواند روی هر نام دانشنامه که مستقیما اورا به صفحه آن دانشنامه میبرد کلیک کند.

# جست و جو

اجازه دهید که کاربر کوئریاش را در کادر جستوجوی نوار کناری تایپ کند و به دنبال یکی از دانشنامه ها بگردد

اگر آن کوئری با نام یکی از دانشنامه ها مطابقت داشت، کاربر باید به صفحه آن مدخل هدایت شود

اگر آن کوئری با نام دانشنامه ها مطابقت نداشت، باید صفحه نتایج جست و جو به کاربر ارائه شود و فهرست تمام دانشنامه ها که آن کوئری زیررشته ای از آنهاست نمایش یابد. مثلاً اگر کوئری جستوجو برابر با ytho باشد، آنگاه Python باید در نتایج جستوجو ظاهر شود .

کلیک روی هر کدام از نامهای مدخل در صفحه نتایج جستوجو باید کاربر را به صفحه آن مدخل ببرد

# صفحه جدید

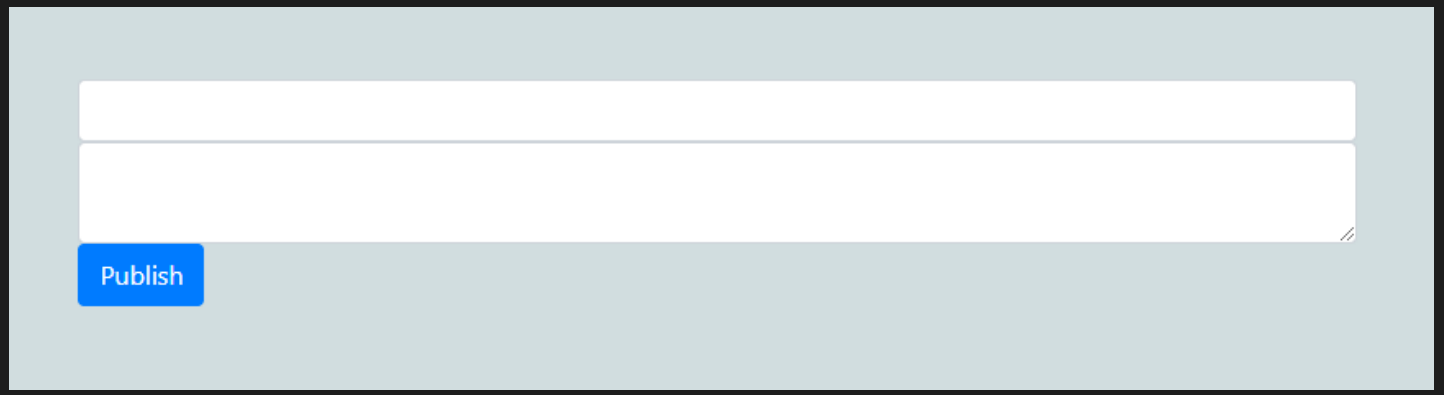
کلیک روی «ایجاد صفحه جدید» در نوار کناری باید کاربر را به صفحه‌ای ببرد که در آنجا بتواند مدخل جدیدی در دانشنامه بسازد

ـ کاربران باید بتوانند عنوانی برای آن صفحه وارد کنند، و در textarea، باید بتوانند محتوای Markdown برای آن صفحه را وارد کنند

ـ کاربران باید بتوانند با کلیک روی دکمه‌ای صفحه جدیدشان را ذخیره کنند

ـ وقتی صفحه ذخیره میشود، اگر دانشنامه، با عنوان داده‌شده، از قبل وجود داشته باشد، باید پیام خطایی به کاربر نشان داده شود

ـ در غیر این صورت، دانشنامه باید داخل دیسک (حافظه) ذخیره شود، و کاربر به صفحه دانشنامه جدید برده شود



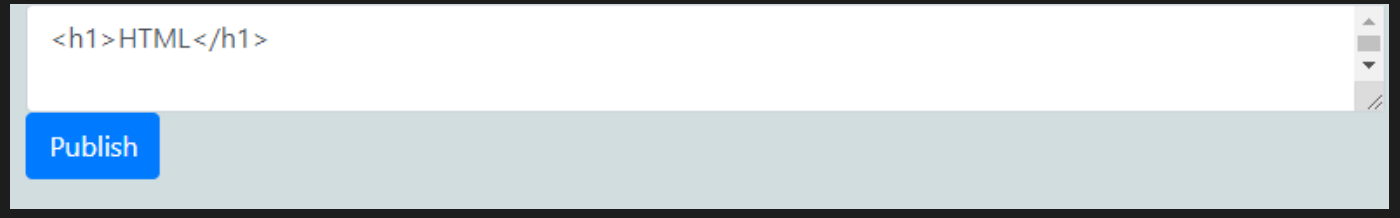
# صفحه ویرایش

در هر صفحه مدخل، کاربر باید بتواند روی لینکی کلیک کند و به صفحهای برود که در آنجا قادر به ویرایش محتوای Markdownآن مدخل در textareaباشد

-Textarea باید از قبل با محتوای Markdownفعلی آن صفحه پر شده باشد (یعنی، محتوای فعلی باید مقدار (valueاولیه textarea)باشد.

-کاربر باید بتواند روی دکمهای کلیک کند و تغییرات صورتگرفته در آن دانشنامه را ذخیره کند .

-به محض ذخیره شدن دانشنامه، کاربر باید دوباره به صفحه آن هدایت شود




## صفحه تصادفی

کلیک روی «صفحه تصادفی» در نوار کناری باید کاربر را به یکی از دانشنامه های تصادفی ببرد.

# تبدیل Markdown به HTML

در هر صفحه دانشنامه، محتوای Markdownدر فایل آن، قبل از نمایش به کاربر، باید به HTMLتبدیل شود. میتوانید از کتابخانه python-markdown2برای این تبدیل استفاده کنید که از طریق pip3 install markdown2قابل نصب است

## راهنمایی

به صورت پیشفرض، با جایگزین کردن یک مقدار در تمپلیت جنگو، جنگو از بخش ، HTMLآن مقدار را نادیده میگیرد تا از خروجی ناخواسته HTMLجلوگیری کند. اگر میخواهید اجازه خروجی رشته HTML را بدهید، میتوانید این کار را با فیلتر safeبکنید (با اضافه کردن |safe بعد از نام متغیری که میخواهید جایگزین کنید

## نحوه ارسال از طریق CS50 ide

https://cs50x.ir/winter/static/web/weeks/submit-web-project.pdf

1-به ide.cs50.io بروید و بر روی "Sign in with GitHub" کلیک کنید تا به CS50 IDE خود دسترسی پیدا کنید.
2-فایل اصلی پروژه خود را درون این دایرکتوری آپلود کنید.
3- cd wikiرا اجرا کنید تا به مسیر پروژه بروید.
4- دستور submit50 web50/projects/2020/x/wiki
را اجرا کنید تا پروژه هفته سوم شما با موفقیت سابمیت شود.

#submit50 از شما نام کاربری اکانت گیت هابتان و Personal Access Token میخواهید که میتوانید طی مراحل ذکر شده نحوه سابمیت کردن (موجود در جلسه صفر ) Personal Access Token خود را دریافت کنید.
ـدر یک ویدیو حداکثر 5 دقیقه ای نحوه کارکرد پروژه خود را نشان دهید و حتما تایم استمپ گذاری کنید .
این فرم را سابمیت کنید

ـ شما میتوانید با رفتن به آدرس : https://cs50.me/cs50wپیشرفت خودتون رو در طول پروژه ببینید.

# نکته ارسال پروژه

وقتی پروژه‌تان را ارسال می‌کنید، محتوای شاخهٔ web50/projects/2020/x/wiki باید با ساختار فایل کد توزیع غیر فشرده، همان‌طور که در ابتدا دریافت شده بود، مطابقت داشته باشد. به عبارتی، فایل‌هایتان نباید داخل هیچ‌کدام از دایرکتوری‌هایی که خودتان ایجاد کرده‌اید بنشینند. شاخهٔ شما همچنین نباید به جز این پروژه، حاوی کد پروژه‌های دیگر باشد. عدم رعایت این ساختار فایل احتمالاً باعث رد شدن کار ارسالی‌تان می‌شود.

برای مثال، در این پروژه، اگر کارمندان نمره‌دهی به آدرس https://github.com/me50/USERNAME/tree/web50/projects/2020/x/wiki بروند (جایی که USERNAME برابر است با نام کاربری‌تان در گیت‌هاب که در فرم زیر ارائه شده است)، باید سه زیر دایرکتوری (encyclopedia، entries، wiki) و همین‌طور فایل manage.py را ببینیم. بعد از بررسی این موضوع، اگر کدتان به این شکل سازماندهی نشده است، مخزن کارهایتان را مطابق با این دیاگرام دوباره ساماندهی کنید.

# CS50x Iran

which uses Harvard University's introduction to the
intellectual enterprises of computer science and the art of programming course

CS50x.ir