



گزارش پروژه کارشناسی

## پیش بینی جای خالی کلمات

استاد پروژه:

جناب آقای دکتر سید کمال الدین غیائی شیرازی

نویسندگان:

محدثه جباری مقدم

فاطمه تبادکانی

گروه آموزشی:

کامپیوتر

پاییز ۱۴۰۰

## تعهدنامه

اینجانب محدثه جباری مقدم و فاطمه تبادکانی دانشجوی دوره کارشناسی رشته مهندسی کامپیوتر دانشکده مهندسی دانشگاه فردوسی مشهد نویسنده پایان نامه پیش‌بینی جای خالی کلمات تحت راهنمایی دکتر غیاثی شیرازی متعهد می‌شوم:

- تحقیقات در این پایان نامه توسط اینجانبان انجام شده و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود و یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه فردوسی مشهد می‌باشد و مقالات مستخرج با نام "دانشگاه فردوسی مشهد" و یا "Ferdowsi University of Mashhad" به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از رساله رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است، اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

### مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه فردوسی مشهد می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

## فهرست مطالب

4	معرفی پروژه .....
4	معرفی Bert .....
5	نشان دادن صفحات سایت .....
5	صفحه‌ی ورود: .....
6	صفحه‌ی ثبت نام: .....
6	اولین صفحه‌ی ورود کاربر: .....
7	اگر کاربر سطح Basic را انتخاب کند: .....
8	اگر کاربر سطح Intermediate را انتخاب کند: .....
9	اگر کاربر سطح Advanced را انتخاب کند: .....
11	کد پیش‌بینی جای‌خالی کلمات: .....
17	توسعه نرم افزاری: .....
17	چرا جنگو را انتخاب کردیم؟ .....
17	نمای کلی از ساختار پروژه: .....
18	مدل‌های دیتابیس: .....
19	فایل View: .....
22	پوشه Template: .....
22	صفحه‌ی ورود و ثبت نام: .....
23	صفحه‌ی جملات: .....
25	فایل Loaddata: .....
27	منابع: .....

## معرفی پروژه

در این پروژه یک متن را به سیستم وارد می‌کنیم و سپس فعل های متن را بدست می‌آوریم و آن ها را به عنوان جای خالی جملات تعریف می‌کنیم.

از Mask-Language Modeling Bert استفاده می‌کنیم و جملات همراه با جای خالی را به مدل می‌دهیم و از Bert می‌خواهیم با توجه به جمله داده شده، جای خالی را برای ما پر کنند. مدل برای هر جای خالی ۳۰۵۲۲ کلمه را پیشنهاد می‌دهد و با توجه به احتمال جواب‌هایی که Bert به ما برمیگرداند، جملات را سطح بندی می‌کنیم.

و در آخر یک اپلیکیشن می‌سازیم که در جهت تقویت و یادگیری زبان از آن می‌توان استفاده کرد به این صورت که جملات و جای خالی ها را به ما نشان می‌دهد و ما باید از بین گزینه های موجود، کلمه ی صحیح را انتخاب کنیم.

جای خالی میتواند برای صفات، قیدها یا اسم ها نیز به کار برود.

## معرفی Bert

Bert یک روش مبتنی بر ترانسفورمرها برای یادگیری بازنمایی زبان است.

این یک bidirectional transformer از پیش آموزش دیده است که به لطف دو رویکرد منحصر به فرد، mask language modeling (MLM) و next sentence prediction (NSP) از موفقیت بی نظیری در NLP برخوردار بوده است.

ما میتوانیم از بازنمایی های زبانی که BERT آموخته است برای کارهای مانند طبقه بندی متن و... استفاده کنیم تا به نتایج پیشرفته ای در مورد مسئله خود دست یابیم.

در بسیاری از موارد، ممکن است بتوانیم مدل bert را که از قبل آموزش داده شده است، خارج کنیم و آن را با توجه به مدل مورد نیاز در مسئله پیاده سازی کنیم.

در این پروژه باتوجه به مسئله fine-tune هم انجام دادیم اما چون دادگان ما انگلیسی است تغییرات چندانی در نتیجه حاصل نشد.

این ترنسفورمر که مورد توجه بسیاری از افراد قرار گرفت.

ما در واقع از Bert اینگونه استفاده می کنیم که یک جمله ناقص وارد می کنیم و از BERT می خواهیم که جمله را برای ما کامل کند.

به عنوان مثال در جمله زیر می خواهیم جای خالی را پر کنیم :

In Autumn the ----- fall from the trees.

در این جمله به احتمال زیاد جواب جای خالی را میدانید و این به این دلیل است که متن جمله را در نظر گرفته اید و مفهوم جمله را متوجه شده اید.

چیزهای زیادی از درختان می افتند به عنوان مثال بلوطها، شاخه ها، برگ ها. اما ما در پاییز شرایط دیگری داریم، که جستجوی ما را محدود می کند، محتمل ترین چیزی که در پاییز از درخت می افتد، برگ است.

Answer: leaves

ما به عنوان انسان ترکیبی از دانش عمومی و درک زبانی برای رسیدن به این نتیجه استفاده می کنیم. برای BERT این حدس از مطالعه زیاد و یادگیری الگوهای زبانی فوق العاده خوب حاصل می شود.

BERT ممکن است نداند پاییز، درختان و برگ ها چیست، اما می داند که با توجه به الگوهای زبانی و زمینه این کلمات، پاسخ به احتمال زیاد برگ ها است.

ما در پروژه یک متن از کتاب Actual test آیلتس را انتخاب کرده ایم.

جای خالی ها را نیز در سه سطح بررسی کردیم: Basic, Intermediate, Advanced:

نشان دادن صفحات سایت

صفحه ی ورود:

## Log In

Username:

Password:

Log In

If you already have no account, [register](#) instead.

## Register

Username:

Email:

Password:

Password confirmation:

If you already have an account, [login](#) instead.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, , pointed toes which help absorb shock when running at high speeds.

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reach its greatest diversity until the recent ice ages, when many of its members became specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and  sheep have been found in arid deserts, while Rocky Mountain sheep survive high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many  s have long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which includes the oryx and the addax, and Antilopinae, which generally contains sligher and more graceful animals such as gazelle and the springbok. The antelope is found in a wide range of habitats, typically woodland, forest, savannah, grassland plains, and marshes. Several species of antelope have adapted to living in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope live in swamps, for instance, the sitatunga has long, splayed hooves that enable it to walk freely and rapidly on swampy ground.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are , each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through . Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when  at high speeds.

Bovids are not so common in endemic insular faunas and are mainly  in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation of 70 new genera. This late Miocene radiation was partly because most bovids became adapted to more open, grassland habitats. Some species of bovid are solitary, but others live in large groups with complex social structures.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is  of a slender, laterally  blade of bone that  from the frontal bones of the skull  a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are

**choose your level**

☐ Basic level

☐ Intermediate level

☐ Advanced level

## اگر کاربر سطح Basic را انتخاب کند:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is Choose here of a slender, laterally Choose here blade of bone that Choose here from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine ( Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

The duiker, Choose here to Cephalophinae sub-family is a small to medium-sized species, Choose here in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, Choose here leaves, shoots, seeds, fruit buds and bark. Some duikers Choose here insects and carrion (dead animal carcasses) from time to time and even Choose here to Choose here rodents or small birds.

Subfamily Caprinae Choose here of mostly medium-sized bovids. Its members are commonly Choose here to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not Choose here its greatest diversity until the recent ice ages, when many of its members Choose here specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and bighorn sheep Choose here Choose here Choose here in arid deserts, while Rocky Mountain sheep Choose here high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically Choose here group. The term is Choose here to Choose here all members of the family Bovidae that do not Choose here under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many antelopes Choose here long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which Choose here the oryx and the addax, and Antilopinae, which generally Choose here slighter and more graceful animals such as gazelle and the springbok. The antelope is Choose here in a wide range of habitats, typically Choose here , forest, savannah, grassland plains, and marshes. Several species of antelope Choose here Choose here to Choose here in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope Choose here in swamps, for instance, the sitatunga has long, splayed hooves that Choose here it to Choose here freely and rapidly on swampy ground.

Bovids are the largest of 10 extant families within Artiodactyla, Choose here of more than 140 extant and 300 species. Fossil evidence five Choose here distinct subfamilies: Bovinae (bison, buffalos, cattle, and relatives), Antelope (addax, oryxes, roan antelopes and relatives), Caprinae (chamois, goats, sheep, and relatives), Cephalophinae (duikers), and

**choose your level**

☒ Basic level

☐ Intermediate level

☐ Advanced level

## کاربر روی اولین جای خالی کلیک میکند:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is Choose here of a slender, laterally Choose here blade of bone that Choose here from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine ( Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

**choose your level**

☒ Basic level

☐ Intermediate level

☐ Advanced level

## گزینه ها را مشاهده می‌کنیم و گزینه‌ی مناسب انتخاب می‌شود.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally Choose here blade of bone that originates from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine ( Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

**choose your level**

☒ Basic level

☐ Intermediate level

☐ Advanced level

اگر کلمه ای که انتخاب می‌شود، درست باشد به رنگ سبز در می‌آید در غیر این صورت به رنگ قرمز در می‌آید.

یک تصویر بالا کلمات یک جای‌خالی دیگر را نمایش داده‌ایم.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally Choose here blade of bone that extends from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine ( Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

choose your level

☒ Basic level

☐ Intermediate level

☐ Advanced level

کلمه‌ی انتخابی دوم صحیح نمی‌باشد.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally Choose here blade of bone that grows from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine ( Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

choose your level

☒ Basic level

☐ Intermediate level

☐ Advanced level

کلمه‌ی grows صحیح می‌باشد.

اگر کاربر سطح Intermediate را انتخاب کند:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are Choose here, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through Choose here. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when Choose here at high speeds.

Bovids are not so common in endemic insular faunas and are mainly Choose here in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation of 70 new genera. This late Miocene radiation was partly because most bovids became adapted to more open, grassland habitats. Some species of bovid are solitary, but others live in large groups with complex social structures.

choose your level

☐ Basic level

☒ Intermediate level

☐ Advanced level

Submit

کلمات برای اولین جای‌خالی نمایش داده می‌شود:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are flexible short branched long, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through Choose here. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when Choose here at high speeds.

choose your level

☐ Basic level

☒ Intermediate level

☐ Advanced level

کلمات برای دومین جای‌خالی نمایش داده می‌شود:



The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are **branched** , each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through **land water running terrain** which help absorb shock when **Choose here** at high speeds.

choose your level

☐ Basic level

☒ Intermediate level

☐ Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are **branched** , each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through **running** which help absorb shock when **Choose here** at high speeds.

choose your level

☐ Basic level

☒ Intermediate level

☐ Advanced level

کلمات صحیح را مشاهده می‌کنیم.

اگر کاربر سطح Advanced را انتخاب کند:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, **Choose here** , pointed toes which help absorb shock when running at high speeds.

choose your level

☐ Basic level

☐ Intermediate level

☒ Advanced level

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reach its greatest diversity until the recent ice ages, when many of its members became specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and **Choose here** sheep have been found in arid deserts, while Rocky Mountain sheep survive high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many **Choose here** s have long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which includes the oryx and the addax, and Antilopinae, which generally contains slither and more graceful animals such as gazelle and the springbok. The antelope is found in a wide range of habitats, typically woodland, forest, savannah, grassland plains, and marshes. Several species of antelope have adapted to living in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope live in swamps, for instance, the sitatunga has long, splayed hooves that enable it to walk freely and rapidly on swampy ground.

Submit

گزینه های مربوط به جای خالی را می‌بینیم:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, **thin slender narrow cushioned sharp** , pointed toes which help absorb shock when running at high speeds.

choose your level

☐ Basic level

☐ Intermediate level

☒ Advanced level

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the

گزینه‌ی صحیح نمایش داده می‌شود:

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, **cushioned**, pointed toes which help absorb shock when running at high speeds.

#### choose your level

- ☐ Basic level
- ☐ Intermediate level
- ☒ Advanced level

## کد پیش‌بینی جای خالی کلمات:

برای بخش هوش مصنوعی این پروژه که در واقع نوعی مسئله NLP (Natural Language Processing) است، همان طور که گفته شده از Bert استفاده کردیم. کد مربوط به این بخش را در ادامه آورده شده و هر بخش توضیح داده می شود:

در این قسمت برای استفاده از کتابخانه های مورد نظر transformers را نصب می کنیم:

```
!pip install transformers

Collecting transformers
  Downloading transformers-4.16.2-py3-none-any.whl (3.5 MB)
    |#####| 3.5 MB 5.2 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from transformers) (21.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface-hub-0.4.0-py3-none-any.whl (67 kB)
    |#####| 67 kB 4.5 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from transformers) (2.23.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (from transformers) (4.62.3)
Collecting sacremoses
  Downloading sacremoses-0.0.47-py2.py3-none-any.whl (895 kB)
    |#####| 895 kB 54.7 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from transformers) (3.4.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (1.21.5)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (2019.12.20)
Collecting tokenizers<0.11.3,>=0.10.1
  Downloading tokenizers-0.11.5-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (6.8 MB)
    |#####| 6.8 MB 30.0 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from transformers) (4.11.0)
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2_12_x86_64.manylinux2010_x86_64.whl (596 kB)
    |#####| 596 kB 62.2 MB/s
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers) (3.10.0.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->transformers) (3.7.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3<1.25.0,1-1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: charset-normalizer<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from sacremoses->transformers) (1.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from sacremoses->transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from sacremoses->transformers) (1.15.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub, transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
  Successfully installed huggingface-hub-0.4.0 pyyaml-6.0 sacremoses-0.0.47 tokenizers-0.11.5 transformers-4.16.2
```

سپس کتابخانه های دیگری که نیاز است را import می کنیم و نسخه pretrain شده bert را دانلود می کنیم:

```
[ ] import numpy as np
import pandas as pd
from transformers import BertTokenizer, BertForMaskedLM
from torch.nn import functional as F
import torch
from random import randrange

[ ] import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger.zip.
True

[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertForMaskedLM.from_pretrained('bert-base-uncased', return_dict = True)

Downloading 100% 28.0/28.0 [00:00<00, 385B/s]
Downloading 100% 229k/229k [00:00<00, 2.91MB/s]
Downloading 100% 459k/459k [00:00<00, 1.94MB/s]
Downloading 100% 570/570 [00:00<00, 11.5kB/s]
Downloading 100% 420M/420M [00:10<00, 42.5MB/s]

Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['cls.seq_relationship.bias', 'cls.seq_relationship.weight']
- This is expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a Bert
- This is NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).
```

به عنوان متن ورودی، از فایل reading.txt استفاده می کنیم. محتویات این فایل را در متغیر main\_text ریخته و “\n” را از آن حذف می کنیم:

```
sample_file = open("/content/drive/MyDrive/reading.txt", "r")
main_text = sample_file.readlines()
sample_file.close()

[ ] main_text

['Bovidae\n',
'A bovid is any member of almost 140 species of ungulates belonging to the family Bovidae. The bovids are the largest family of hoofed mammals and are native to Africa, Europe, Asia, and North America. Members\n',
'\n',
'Bovids are not so common in endemic insular faunas and are mainly recorded in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation\n',
'\n',
'All bovids have the similar basic form-a snout with a blunt end, one or more pairs of horns immediately after the oval or pointed ears, a distinct neck and and a tail varying in length and bushiness among the\n',
'\n',
'Despite differences in size and appearance, bovids are united by the possession of certain common features. Being ruminants, the stomach is composed of four chambers: the rumen (80%), the omasum, the reticulum\n',
'\n',
'Bovids are the largest of 10 extant families within Artiodactyla, consisting of more than 140 extant and 300 extinct species. Fossil evidence suggests five distinct subfamilies: Bovinae (bison, buffalos, cattl\n',
'\n',
'Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animal\n',
'\n',
'Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reac\n',
'\n',
'The duiker, belonging to Cephalophinae sub-family is a small to medium-sized species, brown in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, eating leaves, shoot\n',
'\n',
'The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal b\n']

[ ] for m in main_text:
    if m == '\n':
        main_text.remove(m)

main_text

['Bovidae\n',
'A bovid is any member of almost 140 species of ungulates belonging to the family Bovidae. The bovids are the largest family of hoofed mammals and are native to Africa, Europe, Asia, and North America. Members\n',
'Bovids are not so common in endemic insular faunas and are mainly recorded in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation\n',
'All bovids have the similar basic form-a snout with a blunt end, one or more pairs of horns immediately after the oval or pointed ears, a distinct neck and and a tail varying in length and bushiness among the\n',
'Despite differences in size and appearance, bovids are united by the possession of certain common features. Being ruminants, the stomach is composed of four chambers: the rumen (80%), the omasum, the reticulum\n',
'Bovids are the largest of 10 extant families within Artiodactyla, consisting of more than 140 extant and 300 extinct species. Fossil evidence suggests five distinct subfamilies: Bovinae (bison, buffalos, cattl\n',
'Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animal\n',
'Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reac\n',
'The duiker, belonging to Cephalophinae sub-family is a small to medium-sized species, brown in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, eating leaves, shoot\n',
'The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal b\n']
```

در مرحله بعد در بین جملات موجود به دنبال کلماتی می گردیم که برای جاخالی مناسبتر باشند که برای این کار در هر جمله فعل های آن را انتخاب می کنیم. به این منظور با استفاده از کتابخانه nltk پس از tokenize کردن هر جمله کلماتی که توسط این کتابخانه verb تشخیص داده می شوند را انتخاب کرده و در آرایه verbs ذخیره می کنیم. همچنین با توجه به اینکه افعالی مثل is و are که تعداد حروف کمی دارند معمولاً کلمات آسانی بوده و به عنوان جاخالی مناسب نیستند، افعال با طول کمتر از 3 را در نظر نمی گیریم:

```
[ ] verbs = [[] for i in range(length)]

for i in range(length):
    text = nltk.word_tokenize(main_text[i]);
    pos_tagged = nltk.pos_tag(text)
    for ele in pos_tagged:
        if(ele[1].startswith('VB') and len(ele[0])>3):
            verbs[i].append(ele[0])
```

سپس برای اینکه جاخالی های یک جمله کلمات تکراری نباشند با استفاده از کد زیر کلمات تکراری را از ارایه verbs حذف می کنیم:

```
▶ new_verbs = [[] for i in range(length)]

for i in range(length):
    new_verbs[i] = list(dict.fromkeys(verbs[i]))
```

در قسمت اصلی کد با استفاده از bert, کلماتی که به عنوان جاخالی انتخاب شده اند را در نظر گرفته و کلماتی که Bert برای آن جاخالی ها مناسب می داند را در نظر می گیریم. Bert یک دیکشنری دارد که شامل 30552 کلمه است. پس از ساخت مدل با استفاده از برت، نتیجه کلمات در output ریخته می شود. سپس برای هر کلمه یک احتمال به دست می آورد که در logits ذخیره میشود، این اعداد استاندارد نبوده و ممکن است منفی یا اعشاری یا ... باشند. برای استانداردسازی آن ها از Softmax استفاده می کنیم:

```
for i in range(length):

    for verb in new_verbs[i]:

        labels_one_sentence1 = []
        labels_one_sentence2 = []
        labels_one_sentence3 = []

        split_text = main_text[i].split(verb, 1)
        text1 = split_text[0] + tokenizer.mask_token + split_text[1]

        input = tokenizer.encode_plus(text1, return_tensors = "pt")
        mask_index = torch.where(input["input_ids"][0] == tokenizer.mask_token_id)

        output = model(**input)
        logits = output.logits

        softmax = F.softmax(logits, dim = -1)
        mask_word = softmax[0, mask_index, :]
```

در مرحله بعد کلمات جاخالی را به سه سطح بندی می کنیم. برای این کار باید ببینیم کلمه ای که باید در جای خالی قرار بگیرد، توسط برت به عنوان چندمین کلمه انتخاب می شود. اگر جزء کلمات اول تا ۱۵ ام باشد در سطح ۱ (Basic)، اگر بین ۱۵ و ۲۵۵ امین کلمه باشد سطح ۲ (Intermediate) و اولویت آن بیشتر از ۲۵۶ باشد سطح ۳ (Advanced) قرار میگیرد:

```
level1 = top_10[0:(4**2)-1]
level2 = top_10[4**2:(4**4)-1]
level3 = top_10[(4**4):(4**6)-1]

#level_1
if tokenizer.encode_plus(verb).input_ids[1] in level1:
    answers1[i].append(verb)
    labels_one_sentence1.append(verb)
    options = torch.topk(mask_word, 4, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        if (word!=verb):
            labels_one_sentence1.append(word)
    index = labels_one_sentence1.index(verb)
    correct_index = randrange(4)
    replace(labels_one_sentence1, correct_index, index)
    labels1[i].append(labels_one_sentence1)

#level_2
elif tokenizer.encode_plus(verb).input_ids[1] in level2:
    answers2[i].append(verb)
    labels_one_sentence2.append(verb)
    options = torch.topk(mask_word, 3, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        labels_one_sentence2.append(word)
    correct_index = randrange(4)
    replace(labels_one_sentence2, correct_index, 0)
    labels2[i].append(labels_one_sentence2)

#level_3
elif tokenizer.encode_plus(verb).input_ids[1] in level3:
    answers3[i].append(verb)
    labels_one_sentence3.append(verb)
    options = torch.topk(mask_word, 4, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        labels_one_sentence3.append(word)
    correct_index = randrange(4)
    replace(labels_one_sentence3, correct_index, 0)
    labels3[i].append(labels_one_sentence3)
```

خروجی را در یک dictionary ذخیره می کنیم. فرم آن نیز به این صورت است که برای هر جمله یک دیکشنری ذخیره میشود که شامل text یعنی جمله اصلی، level یعنی سطح آن و آرایه blank که شامل جای خالی هاست، می باشد:



```

dictionary = {"sentences": [{} for i in range(length*3)]}

for i in range(length):
    #level_1
    if(len(answers1[i]) != 0):
        dictionary["sentences"][i].update({"text": main_text[i]})
        dictionary["sentences"][i].update({"level": "Basic"})
        dictionary["sentences"][i].update({"blank": [{} for j in range(len(answers1[i]))]})
        for j in range(len(answers1[i])):
            dictionary["sentences"][i]["blank"][j].update({"correct": answers1[i][j]})
            pos = dictionary["sentences"][i]["text"].index(answers1[i][j])
            dictionary["sentences"][i]["blank"][j].update({"pos": pos})
            dictionary["sentences"][i]["blank"][j].update({"words": labels1[i][j]})
            dictionary["sentences"][i].update({"text": dictionary["sentences"][i]["text"].replace(answers1[i][j], " ", 1)})
    #level_2
    if(len(answers2[i]) != 0):
        k = i+length
        dictionary["sentences"][k].update({"text": main_text[i]})
        dictionary["sentences"][k].update({"level": "Intermediate"})
        dictionary["sentences"][k].update({"blank": [{} for j in range(len(answers2[i]))]})
        for j in range(len(answers2[i])):
            dictionary["sentences"][k]["blank"][j].update({"correct": answers2[i][j]})
            pos = dictionary["sentences"][k]["text"].index(answers2[i][j])
            dictionary["sentences"][k]["blank"][j].update({"pos": pos})
            dictionary["sentences"][k]["blank"][j].update({"words": labels2[i][j]})
            dictionary["sentences"][k].update({"text": dictionary["sentences"][k]["text"].replace(answers2[i][j], " ", 1)})
    #level_3
    if(len(answers3[i]) != 0):
        k = i + length*2
        dictionary["sentences"][k].update({"text": main_text[i]})
        dictionary["sentences"][k].update({"level": "Advanced"})
        dictionary["sentences"][k].update({"blank": [{} for j in range(len(answers3[i]))]})
        for j in range(len(answers3[i])):
            dictionary["sentences"][k]["blank"][j].update({"correct": answers3[i][j]})
            pos = dictionary["sentences"][k]["text"].index(answers3[i][j])
            dictionary["sentences"][k]["blank"][j].update({"pos": pos})

```

سپس این دیکشنری را در یک فایل json ذخیره می کنیم که این فایل ورودی فایل جنگوی ما خواهد بود:

```

import json
json_object = json.dumps(dictionary, indent=len(dictionary))

# Writing to input.json
with open("input.json", "w") as outfile:
    outfile.write(json_object)

```

بخشی از فایل json:





## توسعه نرم افزاری:

در این پروژه برای توسعه نرم افزاری از پایتون استفاده شده است که برای back-end از جنگو و برای front-end از template engine جنگو استفاده کرده ایم.

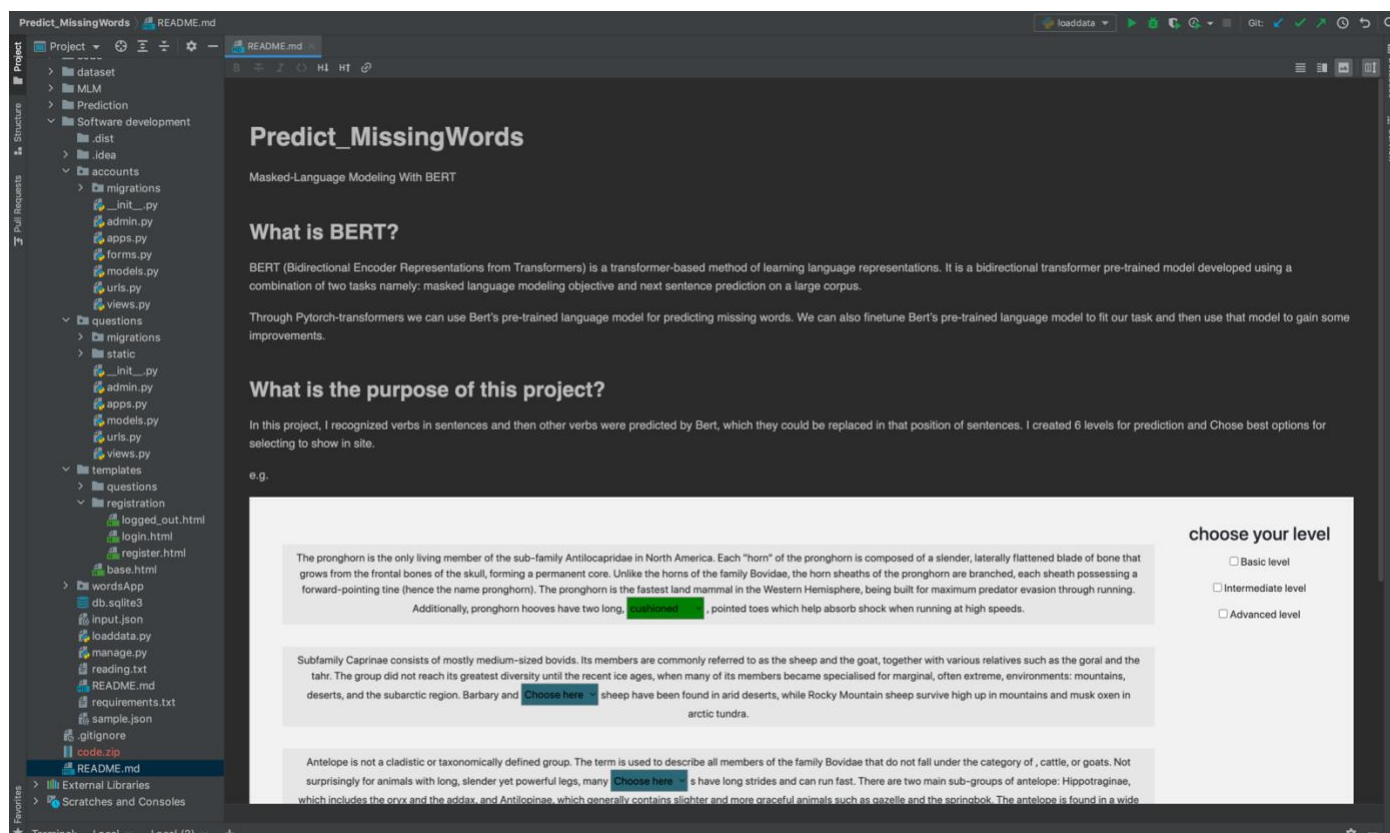
## چرا جنگو را انتخاب کردیم؟

**جنگو (Django)** یک چارچوب نرم افزاری تحت وب آزاد و متن باز است که به زبان پایتون نوشته شده است و از معماری مدل-نما-قالب (Model-View-Template) پیروی می کند. هدف اصلی جنگو ساخت آسان سایت های پیچیده و وابسته به دیتابیس است و بر پایه قابلیت استفاده ی مجدد و قابل اتصال بودن اجزای مختلف، توسعه ی سریع و اصل عدم تکرار طراحی شده است. جنگو سراسر از پایتون استفاده می کند، حتی برای تنظیمات، فایل ها و مدل های اطلاعات.

باتوجه به قابلیت هایی که جنگو در اختیار ما قرار می دهد، تصمیم گرفتیم که از پایتون استفاده کنیم.

## نمای کلی از ساختار پروژه:

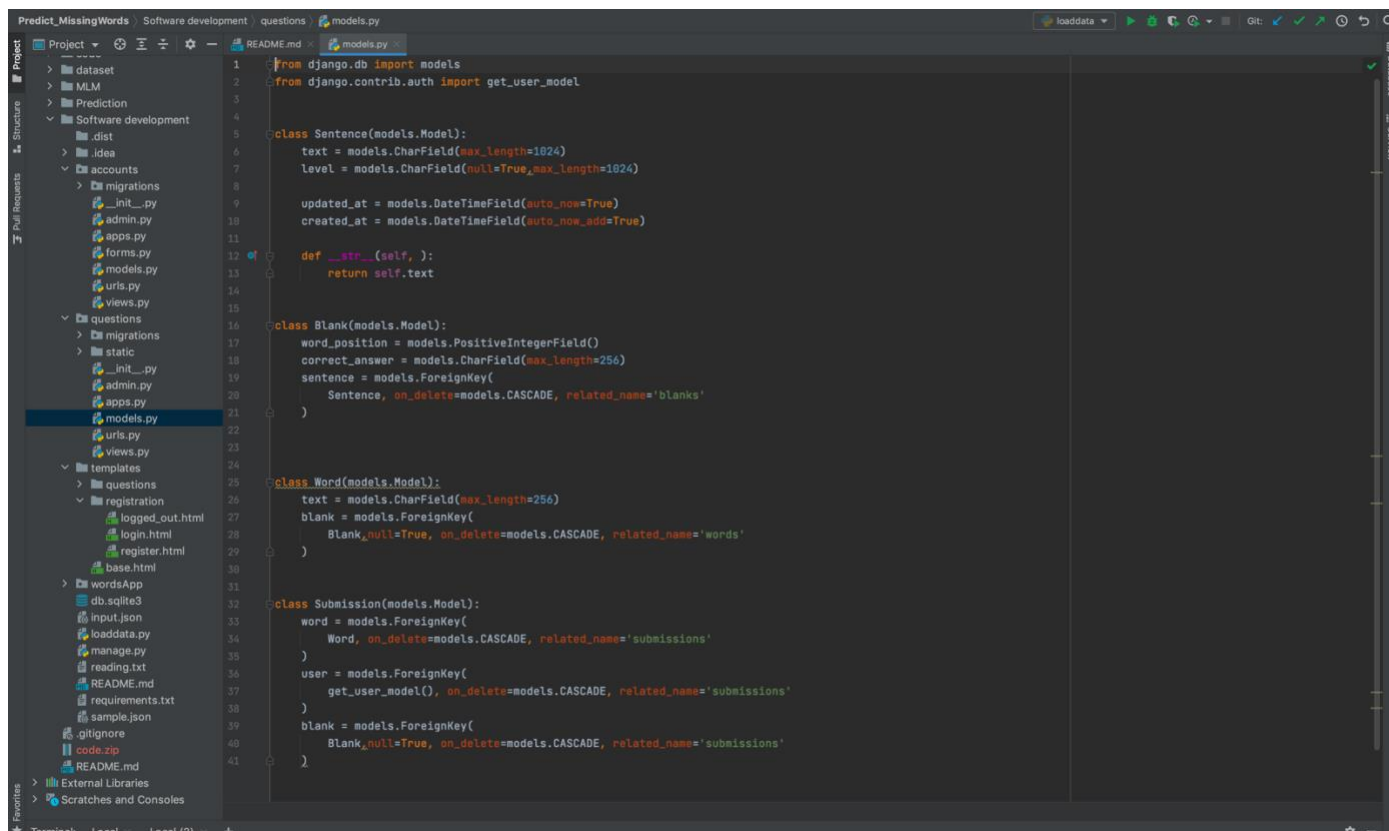
در این صفحه app های مربوط به پروژه رو می بینید که شامل کاربران و سوالات میشود. صفحات manage.py برای ران کردن سرور است و همچنین باقی صفحات را مشاهده میکنید.



## مدل های دیتابیس:

ما در بخش `questions`، در فایل `models.py` دیتابیس را طراحی می‌کنیم. دیتابیس ما یک مدل اصلی به اسم `sentence` دارد که شامل متن و سطح جمله می‌باشد. هر جمله می‌تواند چندین جای خالی داشته باشد که ما آن را با نام `Blank` مشخص کرده ایم و هر جای خالی می‌تواند چندتا کلمه داشته باشد که با نام `word` تعریف شده است.

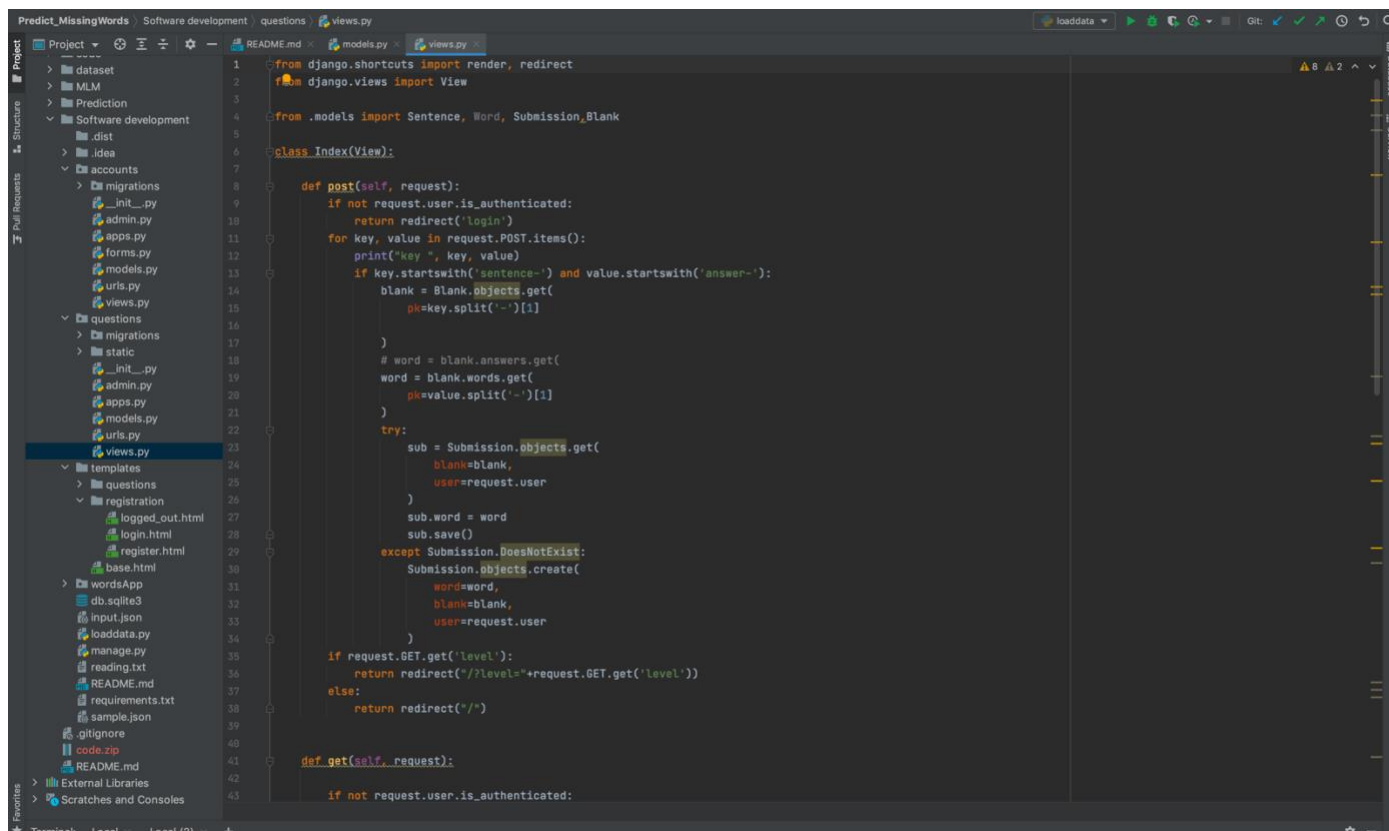
هر کاربر برای هر جای خالی می‌تواند یک گزینه را انتخاب کند که با توجه به همین مسئله اطلاعات کاربر در دیتابیس ذخیره می‌شود.



## فایل View:

در جنگو بخش کنترل کردن کار در فایل view اتفاق می افتد.

در متد post ما submission را انجام می دهیم. با توجه به کلمه ای که کاربر وارد کرده و جای خالی که انتخاب کرده، اطلاعاتش در دیتابیس ذخیره می شود.



```
1 from django.shortcuts import render, redirect
2 from django.views import View
3
4 from .models import Sentence, Word, Submission, Blank
5
6 class Index(View):
7
8     def post(self, request):
9         if not request.user.is_authenticated:
10             return redirect('login')
11         for key, value in request.POST.items():
12             print("key ", key, value)
13             if key.startswith('sentence-') and value.startswith('answer-'):
14                 blank = Blank.objects.get(
15                     pk=key.split('-')[1])
16
17                 # word = blank.answers.get(
18                 word = blank.words.get(
19                     pk=value.split('-')[1])
20                 )
21             try:
22                 sub = Submission.objects.get(
23                     blank=blank,
24                     user=request.user
25                 )
26                 sub.word = word
27                 sub.save()
28             except Submission.DoesNotExist:
29                 Submission.objects.create(
30                     word=word,
31                     blank=blank,
32                     user=request.user
33                 )
34
35         if request.GET.get('level'):
36             return redirect('/?level='+request.GET.get('level'))
37         else:
38             return redirect('/')
39
40     def get(self, request):
41         if not request.user.is_authenticated:
```

در متد get کوئری های لازم زده میشود تا دیتای مورد نظر ما بدست آید و این دیتا را میگیریم و به template engine جنگو میدهیم که html را بگیرد و داخل آن، دیتا را قرار میدهد و به کاربر نشان میدهد.

```

def get(self, request):
    if not request.user.is_authenticated:
        return redirect('login')

    level = request.GET.get('level')

    if level:
        sentences = Sentence.objects.filter(level=level).order_by(
            '-created_at'
        )
    else:
        sentences = Sentence.objects.all().order_by(
            '-created_at'
        )
    sentences_ctx = []

    for sent in sentences:
        current_pos = 0
        blank_data = []
        for blank in sent.blanks.all():
            sub = Submission.objects.filter(
                user=request.user,
                blank=blank
            ).first()

            blank_data.append({
                'id': blank.id,
                'half': sent.text[current_pos:blank.word_position],
                'choices': [
                    {
                        'id': word.id,
                        'word': word.text
                    } for word in blank.words.all()
                ],
                'answer': sub,
                'is_correct': sub.word.text == blank.correct_answer if sub else None
            })
            current_pos = blank.word_position

        sentences_ctx.append({
            "id": sent.id,
            "text": sent.text,
            "level": level,
            "current_pos": current_pos,
            "blank_data": blank_data
        })

    return render(request, 'questions/index.html', {
        'sentences': sentences_ctx,
        'level': level
    })

```

```

def get(self, request):
    if not request.user.is_authenticated:
        return redirect('login')

    level = request.GET.get('level')

    if level:
        sentences = Sentence.objects.filter(level=level).order_by(
            '-created_at'
        )
    else:
        sentences = Sentence.objects.all().order_by(
            '-created_at'
        )
    sentences_ctx = []

    for sent in sentences:
        current_pos = 0
        blank_data = []
        for blank in sent.blanks.all():
            sub = Submission.objects.filter(
                user=request.user,
                blank=blank
            ).first()

            blank_data.append({
                'id': blank.id,
                'half': sent.text[current_pos:blank.word_position],
                'choices': [
                    {
                        'id': word.id,
                        'word': word.text
                    } for word in blank.words.all()
                ],
                'answer': sub,
                'is_correct': sub.word.text == blank.correct_answer if sub else None
            })
            current_pos = blank.word_position

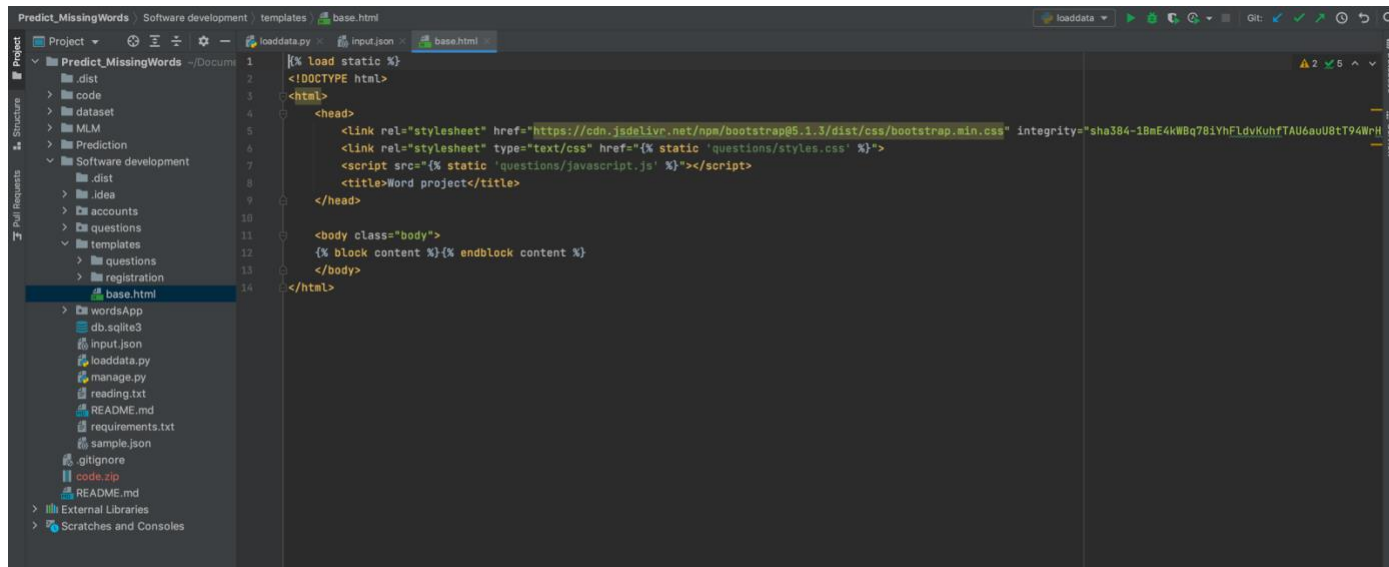
        sentences_ctx.append({
            "id": sent.id,
            "text": sent.text,
            "level": level,
            "current_pos": current_pos,
            "blank_data": blank_data
        })

    return render(request, 'questions/index.html', {
        'sentences': sentences_ctx,
        'level': level
    })

```

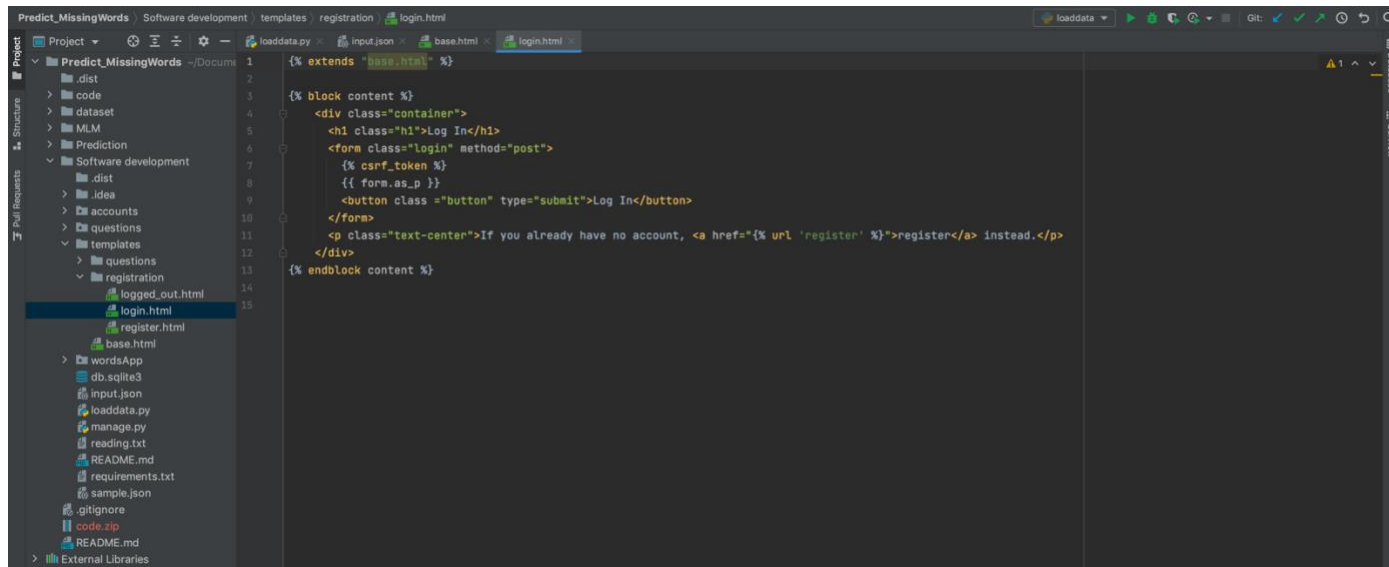
## پوشه Template:

در template فایل های مربوط به html پروژه اینجا تعریف میشوند که یک صفحه ای ابتدایی داریم که لینک css و javascript را داده ایم و باقی صفحات می توانند از آن extend شوند.



```
1  {% load static %}
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq781YhF1dvKuhfTAU6auU8tT94WrH8pJZ" crossorigin="anonymous">
6      <link rel="stylesheet" type="text/css" href="{% static 'questions/styles.css' %}">
7      <script src="{% static 'questions/javascript.js' %}"></script>
8  </head>
9  <body class="body">
10     {% block content %}{% endblock content %}
11 </body>
12 </html>
```

## صفحه ی ورود و ثبت نام:

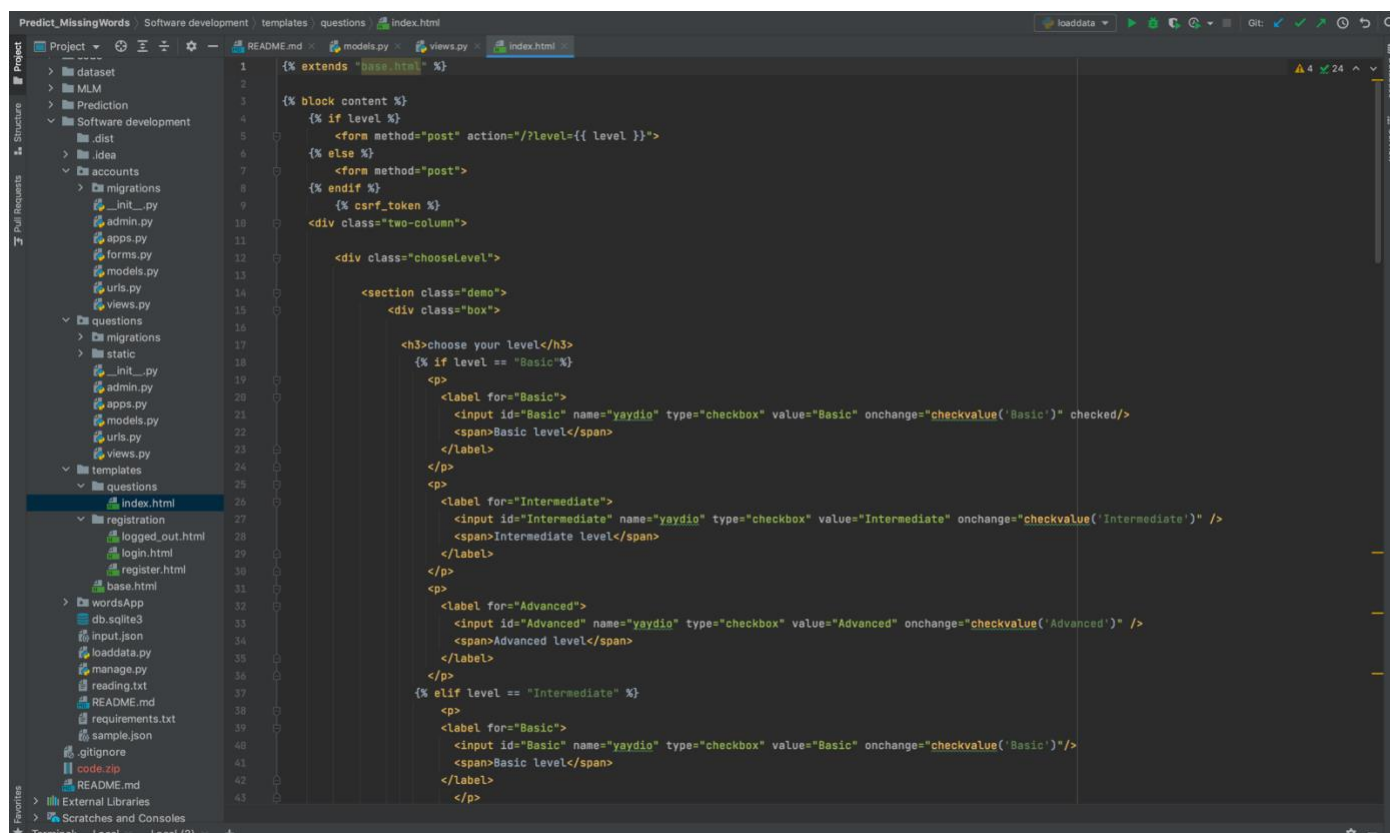


```
1  {% extends "base.html" %}
2
3  {% block content %}
4      <div class="container">
5          <h1 class="h1">Log In</h1>
6          <form class="login" method="post">
7              {% csrf_token %}
8              {{ form.as_p }}
9              <button class="button" type="submit">Log In</button>
10          </form>
11          <p class="text-center">If you already have no account, <a href="{% url 'register' %}">register</a> instead.</p>
12      </div>
13  {% endblock content %}
```

```
1 {% extends "base.html" %}
2
3 {% block content %}
4     <div class="container py-5">
5         <h1 class="h1">Register</h1>
6         <form class="login" method="POST">
7             {% csrf_token %}
8             {{ register_form }}#>
9
10            <div class="two-column row_dir">
11                <span>Username:</span>
12                <div> {{ register_form.username }} </div>
13            </div>
14
15            <div class="two-column row_dir">
16                <span class="text">Email:</span>
17                <div> {{ register_form.email }} </div>
18            </div>
19
20            <div class="two-column row_dir">
21                <span>Password:</span>
22                <div>{{ register_form.password1 }}</div>
23            </div>
24
25            <div class="two-column row_dir">
26                <span>Password confirmation:</span>
27                <div> {{ register_form.password2 }}</div>
28            </div>
29
30            <div>
31                <button class="button" type="submit">Register</button>
32            </div>
33            <p class="text-center">If you already have an account, <a href="{% url 'login' %}">login</a> instead.</p>
34        </div>
35    {% endblock content %}
```

## صفحه‌ی جملات:

در ابتدا چک می‌شود که کاربر سطحی را انتخاب کرده است یا خیر. متناسب با سطحی که کاربر انتخاب می‌کند جملات نشان داده می‌شوند.



```
1 {% extends "base.html" %}
2
3 {% block content %}
4     {% if level %}
5         <form method="post" action="/?level={{ level }}">
6     {% else %}
7         <form method="post">
8     {% endif %}
9         {% csrf_token %}
10        <div class="two-column">
11
12            <div class="chooseLevel">
13
14                <section class="demo">
15                    <div class="box">
16
17                        <h3>choose your level</h3>
18                        {% if level == "Basic"%}
19                            <p>
20                                <label for="Basic">
21                                    <input id="Basic" name="yaydio" type="checkbox" value="Basic" onchange="checkvalue('Basic')" checked/>
22                                    <span>Basic level</span>
23                                </label>
24                            </p>
25                            <p>
26                                <label for="Intermediate">
27                                    <input id="Intermediate" name="yaydio" type="checkbox" value="Intermediate" onchange="checkvalue('Intermediate')" />
28                                    <span>Intermediate level</span>
29                                </label>
30                            </p>
31                            <p>
32                                <label for="Advanced">
33                                    <input id="Advanced" name="yaydio" type="checkbox" value="Advanced" onchange="checkvalue('Advanced')" />
34                                    <span>Advanced level</span>
35                                </label>
36                            </p>
37                        {% elif level == "Intermediate" %}
38                            <p>
39                                <label for="Basic">
40                                    <input id="Basic" name="yaydio" type="checkbox" value="Basic" onchange="checkvalue('Basic')"/>
41                                    <span>Basic level</span>
42                                </label>
43                            </p>
```

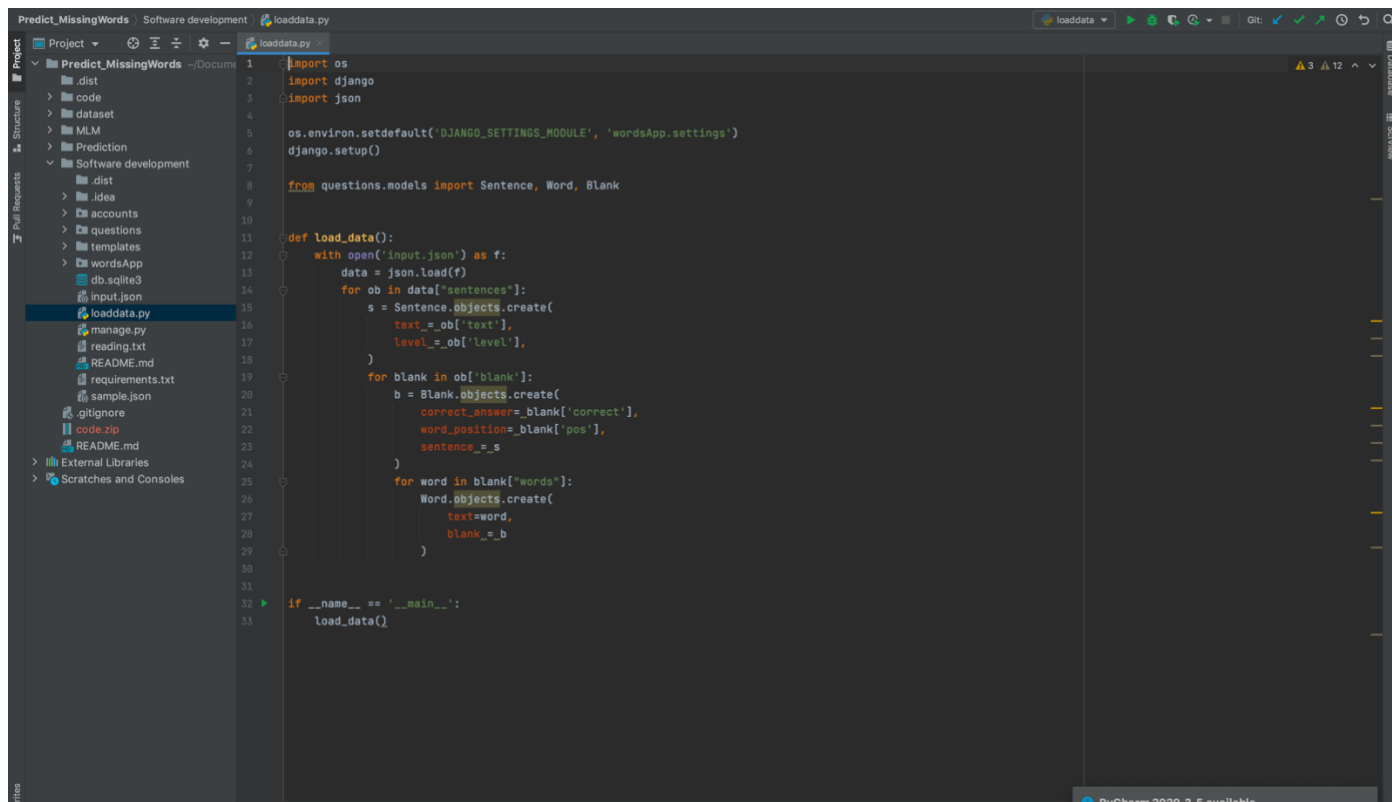
روی تمام جملات در دیتابیس پیمایش صورت میگیرد و روی هر جمله، تمام جای خالی های موجود به کاربر نشان داده می شوند و متناسب با اینکه کاربر چه کلمه ای را برای هر جای خالی انتخاب کند، اگر کلمه ی صحیح باشد رنگ آن سبز می شود و اگر غلط باشد رنگ آن قرمز می شود.



```
103 <div>
104     {% for sentence in sentences %}
105     <div class="question-wrapper">
106         {% for blank in sentence.blanks %}
107         <span class="sentence">{{blank.half}}</span>
108         Has correct answer #}
109         {% if blank.answer and blank.is_correct %}
110         <select class="select-default sel-correct" name="sentence-{{blank.id}}">
111             {% if not blank.answer %}
112             <option value="" selected disabled hidden>Choose here</option>
113             {% endif %}
114             {% for ans in blank.choices %}
115             {% if blank.answer and ans.id == blank.answer.word.id %}
116             <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
117             {% else %}
118             <option class="option" value="answer-{{ans.id}}">{{ans.word}}</option>
119             {% endif %}
120             {% endfor %}
121         </select>
122         Not correct answer #}
123         {% elif blank.answer %}
124         <select class="select-default sel-NotCorrect" name="sentence-{{blank.id}}">
125             {% if not blank.answer %}
126             <option value="" selected disabled hidden>Choose here</option>
127             {% endif %}
128             {% for ans in blank.choices %}
129             {% if blank.answer and ans.id == blank.answer.word.id %}
130             <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
131             {% else %}
132             <option class="option" value="answer-{{ans.id}}">{{ans.word}}</option>
133             {% endif %}
134             {% endfor %}
135         </select>
136         No answer #}
137         {% else %}
138         <select class="select-default" name="sentence-{{blank.id}}">
139             {% if not blank.answer %}
140             <option value="" selected disabled hidden>Choose here</option>
141             {% endif %}
142             {% for ans in blank.choices %}
143             {% if blank.answer and ans.id == blank.answer.word.id %}
144             <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
145             {% else %}
146             <option class="option" value="answer-{{ans.id}}">{{ans.word}}</option>
147             {% endif %}
148             {% endfor %}
149         </select>
150     </div>
151     {% endfor %}
152 </div>
```

## :Loaddata فایل

در این فایل، input.json را میخوانیم و باتوجه به مدل دیتابیزی که طراحی کرده ایم جملات، جای خالی ها و کلمات ذخیره می شوند.



```
1 import os
2 import django
3 import json
4
5 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'wordsApp.settings')
6 django.setup()
7
8 from questions.models import Sentence, Word, Blank
9
10
11 def load_data():
12     with open('input.json') as f:
13         data = json.load(f)
14         for ob in data["sentences"]:
15             s = Sentence.objects.create(
16                 text=_ob['text'],
17                 level=_ob['level'],
18             )
19             for blank in ob["blank"]:
20                 b = Blank.objects.create(
21                     correct_answer=blank['correct'],
22                     word_position=blank['pos'],
23                     sentence=_s
24                 )
25                 for word in blank["words"]:
26                     Word.objects.create(
27                         text=word,
28                         blank=_b
29                     )
30
31
32 if __name__ == '__main__':
33     load_data()
```

برای دیدن کد پروژه می‌توانید به آدرس زیر مراجعه کنید:

[https://github.com/mohadesehjm/Predict\\_MissingWords/](https://github.com/mohadesehjm/Predict_MissingWords/)

<http://jalammar.github.io/illustrated-transformer/>

<https://medium.com/analytics-vidhya/fine-tuning-bert-language-model-to-get-better-results-on-text-classification-3dac5e3c348e>

<https://towardsdatascience.com/masked-language-modelling-with-bert-7d49793e5d2c>

[https://www.tensorflow.org/text/tutorials/fine tune bert](https://www.tensorflow.org/text/tutorials/fine_tune_bert)

<https://github.com/Shivampanwar/Bert-text-classification>