



دانشگاه شیراز

دانشکده مهندسی برق و کامپیوتر

دایکومنت پروژه مهندسی نرم افزار

**Smart Doctor Appointment**

یاسین رضایی

یگانه خواجه پور

حسین میرحسینی

۱۴۰۱

- 1 -

## چکیده

در این پروژه ما به دنبال این هستیم که با استفاده از هوش مصنوعی، بستری را فراهم کنیم که مراجعان و بیماران بتوانند از بیماری خود اطلاع یابند و همچنین به راحتی با پزشک مربوطه لینک شوند. بیمار علائم خود را وارد کرده، بیماری وی گفته می شود و در نهایت لیستی از پزشکان به او پیشنهاد داده می شود که می تواند رزرو کرده و یا با دکتر مورد نظر چت کند.

## فهرست عناوین

صفحه	عنوان
۴	فصل اول: مقدمه
۴	۱.۱ اهداف اسپرینت اول
۴	۱.۲ اهداف اسپرینت دوم
۵	۱.۳ اهداف اسپرینت سوم
۷	فصل دوم: تحقیق و بررسی
۷	۲.۱ برنامه های مشابه
۷	۲.۲ بررسی نیازها
۷	۲.۲.۱ نیازهای کاربر
۸	۲.۲.۲ نیازهای دکتر
۹	فصل سوم: معماری نرم افزار و پیاده سازی
۹	۳.۱ معماری های استفاده شده
۱۱	۳.۲ پیاده سازی
۱۱	۳.۲.۱ سمت سرور
۱۲	۳.۳.۱ سمت موبایل
۱۳	فصل چهار: جزئیات طراحی و پیاده سازی
۱۳	۴.۱ طراحی پایگاه داده
۱۴	۴.۲ طراحی وایرفریم
۱۴	۴.۳ طراحی رابط کاربری
۱۴	۴.۴ هماهنگی تیمی
۱۵	فصل پنجم: دیتاست استفاده شده
۱۷	فصل ششم: ارتباط با پشتیبان
۲۰	فصل هفتم: ایندکس

## فصل اول: مقدمه

امروزه جایی نیست که هوش مصنوعی به گوش ما نخورد. همچنین با پیشرفت روز افزون تکنولوژی، روش های نوین جایگزین روش های سنتی شده اند پس با استفاده از این دو فاکتور به دنبال ارائه خدمتی نوین هستیم. اگر محیطی باشد که بیمار بتواند بدون مراجعه به دکتر و تحمل انواع و اقسام مشکلات، مانند ترافیک و هزینه های زیاد، از بیماری خود اطلاع یابد و در نهایت با دکتر مورد نظر لینک شود، بدون شک رضایت همگان را به عهده خواهد داشت. همچنین پزشکان می توانند در بازاری جدید به دنبال مشتریان جدید باشند و در نهایت به ارتقای آنها منجر شود.

از این رو مجموعه ای از گام ها باید برداشته می شد که در ادامه عنوان می شوند.

### ۱.۱ اهداف اسپرینت اول

اولین گام این بود که بدانیم از اسپرینت اول چه می خواهیم. اهداف ما بدین شکل بودند:

- مشخص کردن معماری درست نرم افزار و ابزارهای استفاده شده
- کسب شناخت از اعضای تیم و مهارت هایشان
- کسب درک متقابل از پروژه و اهداف آن

### ۱.۲ اهداف اسپرینت دوم

پس از اینکه در اسپرینت اول، گروه به هماهنگی رسید، حال باید به سمت پیشبرد اهدافمان حرکت می کردیم، پس در این اسپرینت گام های زیر را در نظر داشتیم:

- پیاده سازی هوش مصنوعی مربوط به پروژه
- پیاده سازی قسمت چت
- بهینه کردن قسمت پروفایل
- کامل کردن قسمت های مورد نیاز برای رابط کاربری

### ۱.۳ اهداف اسپرینت سوم

حال زمان آن بود که پروژه را تکمیل کنیم:

- کامل کردن قسمت تست پزشکی
- کامل کردن قسمت نوبت گرفتن
- تست نهایی
- نوشتن سند نرم افزار

گام بعدی نوشتن user story برای اهدافمان بود که در ادامه عنوان می‌شوند. (برای رعایت قلب مورد نظر اجایل، user story ها به زبان انگلیسی نوشته شده‌اند).

- As a user I want to create an account so I can have my information and test results listed.
- As a user I want to have profile so it can be used as my medical history by doctor.
- As a user I want to be able of changing my information so I get the best out of it anytime.
- As a user I want to be able of taking tests at any time to see what is my illness.
- As a user I want to be able of watching my last tests results so I can have a quick view of my health condition through the time.
- As a user I want to see doctors suggested to me base on my disease.
- As a user I want to see doctor's information such as address, rating and reviews so I can choose the best one.
- As a user I want to be able of making appointments and also canceling it in app so I it saved time and money for me.
- As a user I want to chat with my doctor and get consultation.
- As a user I want to rate doctor and give it a review so others can use it too.
- As a doctor I want to have an account so I can put my information.

- As a doctor I want to be able of changing my information so people can have the latest information about me.
- As a doctor I want to see all my patients' appointments so I can have a precise plan.
- As a doctor I want to be notified if someone makes an appointment or cancel it so I can make plan for it.
- As a doctor I want to see others review and rating of me so I can use it to make my job better.
- As a doctor I want to chat with patients so I can give them consultation online.

## فصل دوم: تحقیق و بررسی

در این بخش به دنبال یافتن نمونه های مشابه و همچنین بررسی نیازهای کاربر و پیاده سازی بودیم.

### ۲.۱ برنامه های مشابه:

برنامه های مشابه پروژه Smart Doctor این برنامه ها بودند:

- [درمانکده](#)
- [محب](#)
- [هوکات](#)

### ۲.۲ بررسی نیازها:

به طور کلی هدف این برنامه، سرعت بخشی به فرایند تشخیص بیماری ها در مراحل اولیه و همچنین ارتباط نزدیک تر با دکترها می باشد. برای همین میتوان نیازها را به دو دسته نیازهای کاربر و همچنین نیازهای دکتر دسته بندی کرد.

#### ۲.۲.۱ نیازهای کاربر:

- امکان تست سریع و قابل اطمینان
- راحتی استفاده از برنامه
- ارتباط سریع و راحت با دکتر مربوطه
- نوبت گرفتن آسان و بدون دردسر

## ۲.۲.۲ نیازهای دکتر:

- اطلاع از زمان های رزرو شده توسط بیماران
- اطلاع از رزروهای کنسل شده
- چت کردن با بیماران در صورت نیاز

## فصل سوم: معماری نرم افزار و پیاده سازی



## ۳.۱ معماری های استفاده شده

معماری استفاده شده MVVM (model-view-viewmodel) می باشد. یک الگوی طراحی معماری نرم افزار است که جداسازی رابط کاربری از قسمت های business logic و data model برنامه را تسهیل می بخشد. بنابراین برنامه ی ما به سه بخش اصلی تقسیم می شود ، سه بخشی که هر کدام می توانند بصورت جدا توسعه داده شوند تا وابستگی بخش ها به هم دیگر کم باشد و همچنین توسعه ی موازی قابل اجرا باشد :

- بخش مدل (Model)

- بخش ویوو (View)

- بخش ویوومدل (ViewModel)

ارتباط بین بخش ظاهر برنامه و منطق برنامه توسط تکنیکی به اسم Data Binding انجام میشود .

مزیت های این معماری :

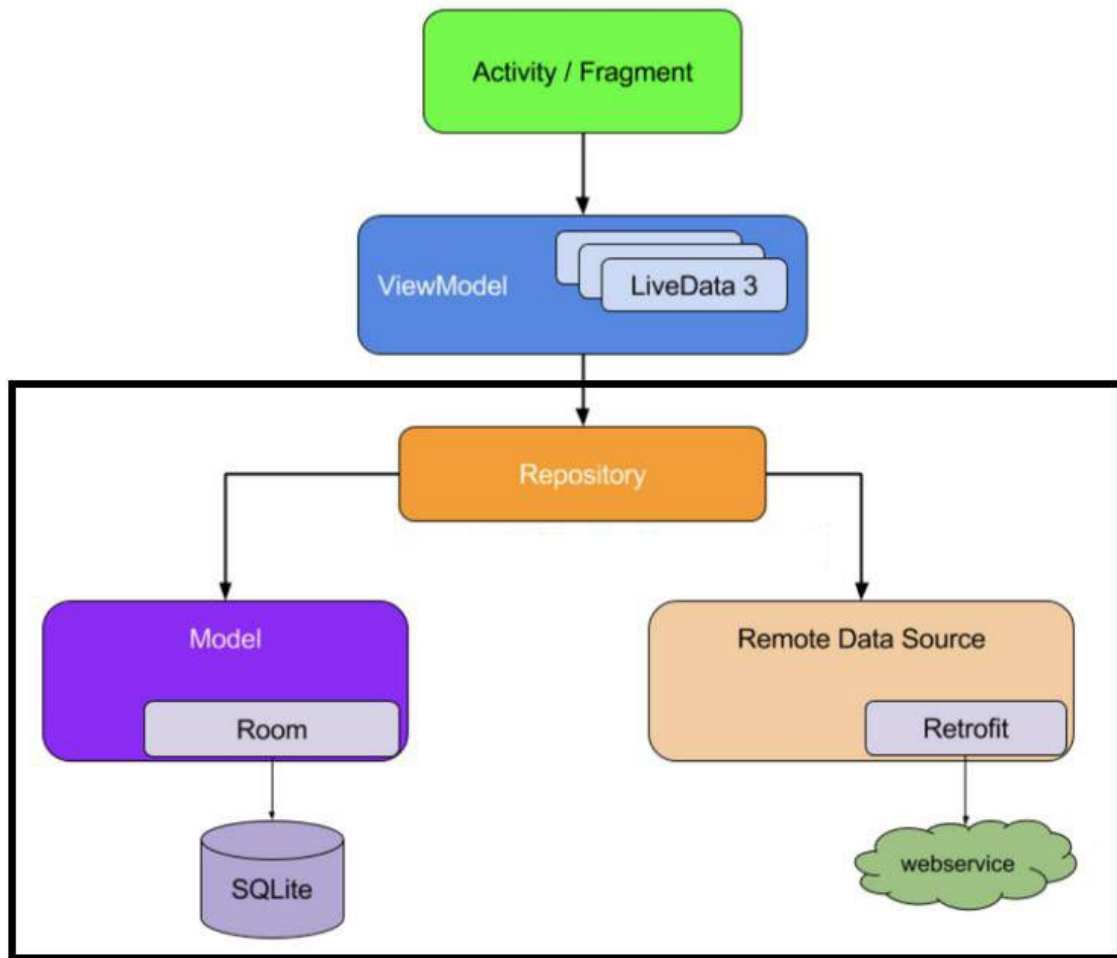
- اجزای رابط کاربری کامل از منطق برنامه جدا است.

- منطق برنامه از عملیات دیتابیسی جدا است.

- خوانش و فهم کد بسیار ساده تر می شود .

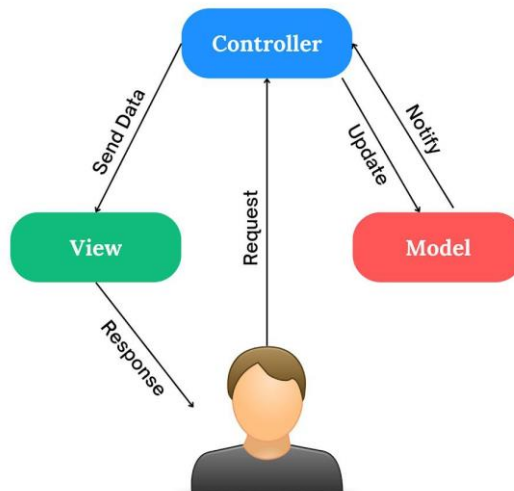
- مدیریت life cycle بسیار آسان تر انجام میشود.

- دلیل اینکه ViewModels وابستگی زیادی به View ندارد نوشتن و انجام تست های گوناگون راحت تر انجام می شود.



همچنین برای سمت سرور از معماری MVC استفاده کردیم. Model View Controller یا به اختصار MVC نوعی روش معماری نرم افزار است که در توسعه وب اپلیکیشن ها بسیار پرکاربرد است.

### MVC Architecture



## ۳.۲ پیاده سازی

پیاده سازی این نرم افزار به دو صورت سمت سرور و سمت موبایل می باشد که در ادامه تکنولوژی های استفاده شده در هر قسمت را عنوان می کنیم.

### ۳.۲.۱ سمت سرور

برای سمت سرور ما از فریمورک جنگو استفاده کردیم. علاوه بر راحتی استفاده و همچنین امکان پیاده سازی دیتابیس با کمک model Django، دلیل دیگر استفاده از این فریمورک پیاده سازی مدل هوش مصنوعی برای تست پزشکی بود که در آنجا با استفاده از دیتاست مربوط به سایت kaggle، یک مدل یادگیری ماشین را با استفاد از روش Decision Tree بود. در پیاده سازی هوش مصنوعی نرم افزار پس از خواندن دیتایی با ۱۳۲ ستون و بیش از ۴۰۰۰ سطر توانستیم مدلی با کارایی ۱۰۰ درصد ارائه دهیم، که در ریپازیتوری مربوط به پروژه و پوشه Machine Learning موجود می باشد.

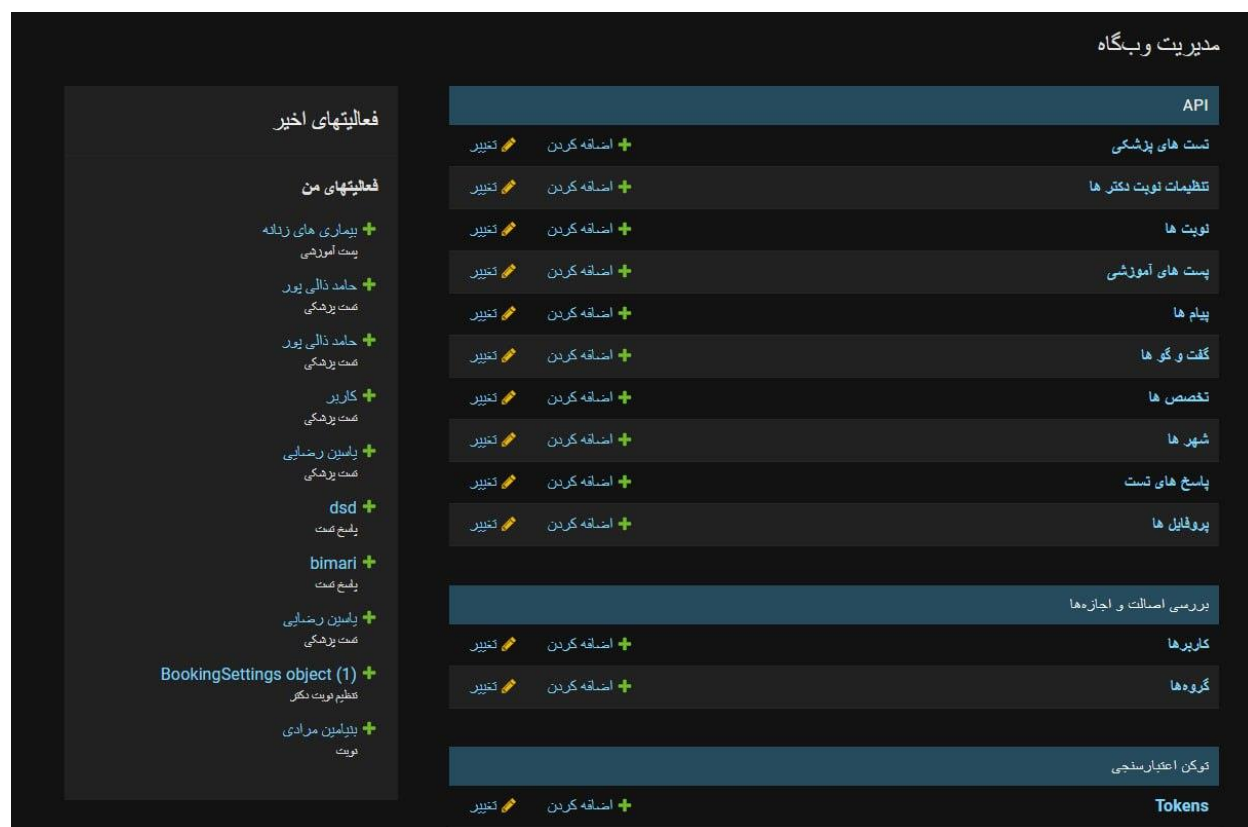
### ۳.۲.۲ سمت موبایل

- Kotlin: برای پیاده سازی نرم افزار به صورت اپ اندروید که راحتی استفاده را به همراه داشته باشد.
- Retrofit: برای فراخوانی api ها از سمت سرور
- XML: برای طراحی رابط کاربری نرم افزار
- Dagger Hilt: برای مدیریت Dependency Injection در نرم افزار اندرویدی

## فصل چهار: جزئیات طراحی و پیاده سازی

### ۴.۱ طراحی پایگاه داده

از مهمترین قسمت های هر پروژه، طراحی پایگاه داده مربوطه و همچنین نیازهای مرتبط با آن می باشد. از این روی پس از طراحی ERD که در فصل آخر گزارش قابل مشاهده است، با استفاده از مدل جنگو، آن را پیاده سازی کردیم. در ادامه تصویری از پنل مدیریت پایگاه داده را مشاهده می کنید.



## ۴.۲ طراحی وایرفریم

بعد از مشخص کردن اهداف اسپرینت اول و همچنین user story، باید نمایی از آنچه کاربر در برنامه انتظار دارد تهیه می شود. از این رو با طراحی wireframe اولین قدم های طراحی برنامه را برداشتیم. فایل مربوط به طراحی وایرفریم در فصل آخر موجود می باشد.

## ۴.۳ طراحی رابط کاربری

پیش از طراحی بخش Front-end توسط XML، رابط کاربری را در نرم افزار Figma طراحی می کردیم تا به کمک آن بتوانیم با دید بهتری پیاده سازی بکنیم. لینک مربوطه به طراحی های Figma در فصل آخر موجود می باشد.

## ۴.۴ هماهنگی تیمی

بعد از اینکه توانستیم توانایی های افراد را شناسایی کنیم، به صورت مداوم و دوره ای جلسه هایی تحت عنوان daily standup داشتیم و خروجی و مباحث مطرح شده هرکدام را صورت جلسه کردیم. اطلاعات مربوط به هر daily standup و خروجی آن ها در فصل آخر قابل مشاهده است.

همچنین برای هماهنگی در طول پیاده سازی، تمامی کدهای بر روی ریپازیتوری گیتهاب به صورت مداوم commit می شد که همین مورد نیز در امر deploy نیز به ما کمک می کرد بدین شکل که Liara هنگام deploy تمامی کامیت های ریپازیتوری ما را نیز در نظر می گرفت. لینک ریپازیتوری عنوان شده نیز در فصل هفتم موجود می باشد.

## فصل پنج: دیتاست استفاده شده

دیتا در دنیای هوش مصنوعی، همانند سوخت برای اتوموبیل است. بدون دیتا هیچ یادگیری برای ماشین وجود ندارد. پس اولین قدم برای پیاده سازی مدل هوش مصنوعی مدنظر، پیدا کردن دیتاست مناسب بود. پس از جستجوهای فراوان دیتاستی با ۱۳۳ ستون و نزدیک به ۵۰۰۰ سطر پیدا کردیم.

در ۱۳۲ ستون ابتدایی، علائم فرد بیماری وجود داشتند. از علائم پوستی مانند خارش و تاول گرفته تا ضعف عضلانی همه موجود بودند. سپس در ستون آخر بر اساس این علائم بیماری شخص گفته میشد که این دیتاست قادر به تشخیص ۴۱ بیماری منحصر به فرد بود.

coma	stomach_t	distention	history_of	fluid_over	blood_in_	prominent	palpitation	painful_wi	pus_filled	blackhead	scurrying	skin_peelii	silver_like	small_deni	inflammat	blister	red_sore_	yellow_cri	prognosis
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 GERD
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 GERD
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 GERD

نمایی از دیتاست استفاده شده

پس از اینکه دیتاست مورد نظر را بررسی کردیم، برای راحتی استفاده هر مورد را به فارسی ترجمه کردیم و آنها را در دیتاست های متفاوت برای خواندن و کوئری گرفتن قرار دادیم.

سپس چون تعداد علائم بیماری خیلی زیاد بود و اگر قرار به انتخاب از بین ۱۳۲ مورد بود خیلی سخت می‌شد، هرکدام را در دسته های جدا قرار دادیم تا پیاده سازی آن در سمت موبایل راحت تر باشد و همچنین اصول user friendly بودن نرم افزار را رعایت کرده باشیم.

فارسی	English	Section
خارش	itching	پوست
کهیر	skin_rash	پوست
تاول پوستی	nodal_skin_eruptions	پوست
زردی پوست	yellowish_skin	پوست
لکه های قرمز روی بدن	red_spots_over_body	پوست
لکه های پوستی	dyschromic patches	پوست
جوش های چرکی	pus_filled_pimples	پوست
جوش های سر سیاه	blackheads	پوست
پوست پوست شدن	skin_peeling	پوست
خاکستری شدن رنگ پوست	silver_like_dusting	پوست
تاول	blister	پوست
ترشح پوست زرد	yellow_crust_ooze	پوست
سختی حرکت کردن	movement_stiffness	حرکتی
حرکات چرخشی	spinning_movements	حرکتی
عدم تعادل	loss_of_balance	حرکتی
بی ثباتی	unsteadiness	حرکتی
ضعف یک سمت بدن	weakness_of_one_body_side	حرکتی
سردی دست و پا	cold_hands_and_feets	دست و پا
ضعف دست و پا	weakness_in_limbs	دست و پا

نمونه ای از دسته بندی علائم بیماری

در نهایت بعد از تمامی این کارها، هر بیماری را به تخصص پزشکی مربوطه مرتبط کردیم تا به بیماران و کاربران، دکتری با تخصص مربوطه پیشنهاد شود.

Desease	Persion	Doctor
Fungal infection	عفونت قارچی	پوست
Allergy	آلرژی	عمومی
GERD	رفلاکس معده به مری	گوارش
Chronic cholestasis	کلستاز مزمن ( کبدی )	داخلی
Drug Reaction	حساسیت دارویی	عمومی
Peptic ulcer disease	زخم معده	گوارش

نمونه ای از دسته بندی تخصص های پزشکی




## فصل شش: ارتباط با پشتیبان

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي كَبَدٍ (سوره بلد، آیه ۴): که ما انسان را به حقیقت در رنج و مشقت آفریدیم!

بدون شک طی کردن هر فرآیند بزرگی نیازمند ارتباط مستقیم با عوامل دخیل در آن فرآیند می باشد. در کنار ارتباط با product owner برای یافتن درک متقابل در رابطه با کم و چون پروژه، تیم ما نیز ارتباط تنگاتنگی با پشتیبانی Liara در زمینه deploy داشت!

گفت و گوهایی که عموماً به خاطر عواملی بودند که ما را به سکتۀ مغزی نزدیک می کردند! در ادامه تصاویری از گفت و گوهای صورت گرفته که اکثراً مربوط به روزهای آخر می شوند را مشاهده می کنید. صحبت هایی که پشت هر کدام استرس زیاد، فشار خون بالا و همچنین احتمال سکتۀ مغزی پنهان بود!

<p>لطفا پکیج <code>mysqldclient</code> رو در فایل <code>requirements.txt</code> قرار داده و مجددا برنامه‌تون رو دیپلوی بفرمایید.</p>	 محمدرضا واشیان سه‌شنبه ۴ بهمن (۱۶:۰۷)
<p>الان دیپلوی کردم آخرین ورژن رو</p>	 یاسین رضایی سه‌شنبه ۴ بهمن (۱۵:۵۹)
<p>در صورت امکان لطفا مجددا آخرین ورژن برنامه‌تون رو دیپلوی بفرمایید و در همین تیکت اطلاع بدید که همکاران فنی این مورد رو بررسی کنند.</p>	 محمدرضا واشیان سه‌شنبه ۴ بهمن (۱۵:۵۶)
<p>برنامک با کد 255 پایان یافت. (non-zero exit) چنین خطایی بهم داده میشه</p>	 یاسین رضایی سه‌شنبه ۴ بهمن (۱۵:۴۹)
<p>سلام پس از استقرار برنامه دچار چند بار خطای پشت سر هم میشه و درنهایت خاموش میشه کلا خطای خاصی در لاگ نشون نمیدع</p>	 یاسین رضایی سه‌شنبه ۴ بهمن (۱۵:۴۵)
<p>سلام آقای رضایی. وقت بخیر لطفا درمورثی که با خطای به‌خصوصی مواجه هستید. اسکرین‌شات خطا رو از بخش پیوست‌های این تیکت برای ما ارسال کنید تا بتونیم بهتر در رفع این مورد شما رو راهنمایی کنیم.</p>	 محمدرضا واشیان سه‌شنبه ۴ بهمن (۱۵:۴۴)
<p>فعلا برگشتم به ورژن های قبلی و اوکی کار میکنه اما ممنون میشم اگه مشکل ورژن های 30 تا 34 رو متوجه شدید بهم بگید چون خیلی ضروریه</p>	 یاسین رضایی سه‌شنبه ۴ بهمن (۱۵:۳۷)
<p>سلام پس از دیپلوی اخرم برنامه به درستی اجرا نمیشه و من هم نمیدونم مشکل چیه ممنون میشم راهنمایی کنید</p>	 یاسین رضایی سه‌شنبه ۴ بهمن (۱۵:۰۸)



یاسین رضایی

چهارشنبه ۵ بهمن (۱۱:۰۱)


بله درست شد

اصلا حواسم به این موضوع لینوکسی بودن نبود

ممنون از شما

خیلی لطف کردید

---



محمدرضا واشیان


چهارشنبه ۵ بهمن (۱۰:۴۸)

در نظر داشته باشید سرورهای لیا را لینوکسی هستند نیاز هست آدرس دهی رو متناسب با این OS انجام بدید.

لطفا این مسیر رو جایگزاری کرده و مجددا بررسی بفرمایید مشکل رفع شده یا خیر:

```
df - pd.read_csv("api/TestModel/training_data.csv")
```

---



یاسین رضایی

چهارشنبه ۵ بهمن (۱۰:۳۸)

سلام

ممنون بابت پاسخگویی


من داخل اپلیکیشن api و پوشه TestModel یک فایل csv دارم

و توی همون دایرکتوری داخل فایل ML.py یکی از فانکشن های این فایل به اسم mlFunction رو وقتی میخوانم اجرا کنم این خطای نبود نبود فایل یا دایرکتوری رو بهم بده

روی لوکال الان برام درست کار میکنه اما بعد از دیپلوی به مشکل میخورم

SCREENSHOT 2023-01-25 103501.PNG

---



محمدرضا واشیان

چهارشنبه ۵ بهمن (۱۰:۱۹)


سلام آقای رضایی، وقت بخیر

لطفا به صورت relative path داخل `api/Data analysis/main.py` عمل کرده و بررسی بفرمایید مشکل حل میشه یا خیر.

در صورتی که مشکل حل نشد، لطف بفرمایید داخل چه مسیری از برنامه با این خطا مواجه میشد تا بهتر بتوانم راهنمایی‌تون کنم.

در خدمتیم.

---



یاسین رضایی

سه‌شنبه ۴ بهمن (۲۳:۰۹)

با سلام من به دایرکتوری برای به فایل csv داخل فایل هام دارم که به صورت لوکال برنامه اونو درست پیدا میکنه

به همچین دایرکتوری

`api\TestModel\training_data.csv`

اما وقتی دیپلوی میکنم با ارور زیر مواجه میشم

`'No such file or directory: 'api\TestModel\training_data.csv [Errno 2]`

ممنون میشم راهنمایی کنید ؛)

پیشاپیش ممنونم از وقتی که میزاید

## فصل هفت: ایندکس

در این فصل تمامی فایل های مورد نیاز را قرار داده ایم.

- [وایر فریم](#)
- [ERD](#)
- [خلاصه جلسات \(Daily standup\)](#)
- [رابط گرافیکی در فیگما](#)
- [فایل ویدیو تست](#)
- [ریپازیتوری گیت هاب](#)



**Shiraz University**

## **Smart Doctor Appointment**

**Yasin Rezaei**

**Yeganeh Khajepoor**

**Hosein Mirhoseini**

**Fall 2022**