

به نام خدا



داک امکان سنجی ( Study Feasibility )

**اعضای گروه:**

مبین آزادانی

علیرضا دوستی مهر

محمدجواد جلیلود

مرتضی ملکی نژاد شوشتری

رضا یزدی

دانیال یگانه

# فهرست مطالب

۱	معرفی پروژه	۳
۲	جامعه هدف	۳
۳	محصولات مشابه در بازار	۳
۴	اسکوپ پروژه	۴
۵	تکنولوژی مورد استفاده	۵
۶	برنامه تجاری	۵
۷	ریسک ها	۵
۱.۷	منابع انسانی	۵
۲.۷	ریسک های فنی	۶
۳.۷	ریسک های امنیتی	۶
۴.۷	رقبای محصول	۷
۵.۷	بازاریابی	۷

## ۱ معرفی پروژه

در جامعه برنامه نویسان یکی از مهم ترین راه های یادگیری حل مسائل متنوع و دیدن راه های مختلف برای حل یک سوال است. علاوه بر این بسیاری از شرکت ها برای استخدام با داوطلبان زیادی مواجه هستند که طی کردن فرآیند مصاحبه برای هرنفر فرآیندی طولانی و هزینه بر است. ما تصمیم گرفتیم که یک سیستم داوری سوالات برنامه نویسی طراحی کنیم که هم به برنامه نویس ها برای حل سوال کمک کند و هم این امکان را داشته باشد که بتوان در آن مسابقه طراحی کرد و به این طریق شرکت ها بتوانند قبل از فرآیند مصاحبه و یا چک کردن رزومه معیار سنجش راحت تری برای انتخاب نیروهای جدید داشته باشند. از اهداف این پروژه میتوان به این موارد اشاره کرد (فیچرست کامل پروژه در بخش اسکوپ پروژه آمده).

- امکان ساخت راحت مسابقه

- امکان تمرین و حل سوالات متعدد

- افزایش یادگیری استفاده کننده برنامه به وسیله آموزش هایی که میتوان در کنار سوالات قرار داد

- امکان تست جواب و در صورت خطا دیدن دلیل آن

## ۲ جامعه هدف

جامعه هدف این برنامه چند گروه عمده هستند. اول برنامه نویسانی که به دنبال یافتن و حل مسئله برای رشد توانایی خود هستند. دوم دانشگاه ها، مراکز آموزشی و شرکت هایی که به دلایل مختلفی به پلتفرمی برای برگزاری مسابقه برای سنجش توانایی برنامه نویسان نیاز دارند و سوم برنامه نویسانی که به دنبال کار میگردند و از این راه می توانند با شرکت مورد نظر خود ارتباط برقرار کنند.

از آنجایی که امکان طرح سوال با سختی های مختلفی وجود دارد، این برنامه محدودیت خاصی بر اساس توانایی افراد ندارد و تنها شاید سوالات مربوط به یک سطح بیشتر باشند، همچنین از آنجا که در بخش سوالات (و نه در مسابقه) یک بخش آموزشی قرار دارد امکان حل سوال هایی که کمی سخت تر هستند و با راهنمایی کمی حل می شوند نیز وجود دارد.

## ۳ محصولات مشابه در بازار

قبل از اینکه به محصولات مشابه پرداخته شود، نکته ای که باید توجه شود این است که از آنجا که جامعه هدف این نوع برنامه ها عموم نیستند پس تعداد کاربر نیز باید در حیطه خودش مقایسه شود.

۱. Codeforces

موفقیت ها:

- دارای بیش از ۶۰۰۰۰۰ کاربر

- یکی از مهم ترین سایت ها در زمینه programming competitive

- بلاگ کدفرسز تبدیل به یک محل گفتگو برای programmer competitive ها شده

• ارتباط با دانشگاه ها و شرکت ها از طریق برگزاری مسابقات و بخش gym سایت

ویژگی ها:

- برگزاری منظم مسابقات در سطوح مختلف
- وجود tutorial در کنار سوالات در بخش problemset
- وجود بخش gym برای کلاس های مختلف
- وجود امکان hack بعد از مسابقه برای به چالش کشیدن کد سایر شرکت کنندگان
- سیستم ریتینگ بر اساس عملکرد شرکت کنندگان در مسابقه
- بلاگ فنی که در آن کاربران مختلف میتوانند مطلب بنویسند

۲. Quera

موفقیت ها

- به عنوان معروف ترین سیستم داوری ایرانی، کوئرا با بسیاری از شرکت ها همکاری داشته
- بسیاری از دانشگاه های سطح بالای کشور از کوئرا برای اهداف آموزشی استفاده می کنند
- کوئرا دوره ها و رویداد های مختلفی برگزار کرده تا هم به کسب درآمد بهتری برسد و هم جامعه خود را غنی تر کند

ویژگی ها

- تمرکز زیاد بر روی بخش BYB
- برگزاری دوره های آموزشی مختلف بعضا با همکاری شرکت های مختلف
- علاوه بر سوال های الگوریتمی، کوئرا از سوال های تکنولوژی و ... نیز پشتیبانی و سیستم داوری آن بر روی تعدادی از آنها نیز کار میکند
- امکان تعریف جواب هم به صورت ثابت و هم به صورت داینامیک

## ۴ اسکوپ پروژه

۱. بخش authentication و user management

- امکان ثبت نام کاربر جدید و ورود کاربر قدیم
- امکان تغییر اطلاعاتی مثل نام، شهر، ایمیل، شماره تلفن، عکس پروفایل و ...

۲. بخش سوالات

- امکان طرح سوال جدید و حذف سوال قدیمی که توسط کاربر طراحی شده
- دیدن صورت سوال و توضیحات دیگر مثل نمونه ورودی، خروجی
- امکان سابمیت کردن کد برای تست درست بودن جواب و داوری درست آن (در حال حاضر ما در نظر گرفتیم تنها از یک زبان پشتیبانی کنیم به دلیل چالش های امنیتی مختلف آن)

- تشخیص ارور های مختلف برنامه مثل جواب غلط، خطاهای مختلف، ، مصرف زیاد مموری یا طول کشیدن بیش از حد آن و گزارش به کاربر
- مشاهده بخش آموزشی سوال (در صورت وجود)

### ۳. بخش مسابقه

- تعریف مسابقه و اضافه کردن سوالات مختلف به آن
- ست کردن محدودیت زمانی روی مسابقه
- امکان شرکت در مسابقه توسط کاربران و اجرای سیستم داوری روی کد ها
- مشاهده جدول امتیازات مسابقه

## ۵ تکنولوژی مورد استفاده

Backend: Golang  
Storage: Postgres, Redis, Minio,  
Broker: Nats  
Infrastructure: docker compose  
Frontend: React

## ۶ برنامه تجاری

- بازاریابی به نظر ما بهترین برنامه برای بازاریابی استفاده برنامه در دانشگاه های مختلف، و انتشار آن در جاهای مختلف است.

یک مسئله دیگری که وجود دارد تعداد کم سیستم های داوری کد آزاد و متن باز است به این طریق برنامه ما میتواند در محیط هایی که نیاز به یک سیستم داوری سوال برای خود دارند هم استفاده شود (بر خلاف مثلا کوئرا که مسابقه درون خود سیستم برگزار می شود)

- روش کسب درآمد

شیوه های مختلفی برای کسب درآمد وجود دارد، یکی تبلیغات است که ما زیاد روی آن اهمیت ندادیم برنامه اصلی ما برای کسب درآمد پلن های B2B است که شرکت هایی که میخواهند از پلتفرم ما استفاده کنند با توجه به ابعاد و نوع مسابقه قرارداد ببندند و همینطور میتوان از ایونت ها و ... برای کسب درآمد نیز استفاده کرد.

## ۷ ریسک ها

### ۱.۷ منابع انسانی

ریسک های منابع انسانی در اوایل پروژه به نسبت کمتر هستند ولی باز مثال هایی مثل حذف درس، بروز اختلاف، بروز مشکل برای یک از اعضا و .... وجود دارد.

راهی که ما اینجا در نظر گرفتیم این بود که در اسکوپ پروژه کمی فیچر ها را کمتر از حدی که به نظر خود میتوانیم در بهترین شرایط در بهترین حالت برسیم در نظر گرفتیم و یک سری از فیچر ها را به فیچرست آپشنال انتقال دادیم. یکی دیگر از راه هایی که ما در نظر گرفتیم این است که از وجود siloos knowledge جلوگیری کنیم و اینطور نباشد که یک بخش از پروژه فقط توسط یک نفر قابل درک باشد. در غیر اینصورت با خروج یک نفر از تیم زمانی باید صرف آنبردینگ بقیه روی آن تکه کد شود.

ریسک های منابع انسانی بعد از ترم اگر بخواهیم پروژه را ادامه بدهیم به نسبت بیشتر است چون ممکن است افراد حس مسئولیت پذیری کمتری کنند، یکی از دلایلی که ما تصمیم گرفتیم از گواهی GPL برای این برنامه استفاده کنیم همین مورد است که سایر افراد خارج از تیم نیز بتوانند روی آن contribute کنند.

## ۲.۷ ریسک های فنی

موارد مهمی که برای این مورد به ذهن ما رسید این موارد است:

- obsolete شدن کتابخانه ها و ابزار های ما
  - طولانی شدن تعداد کد های داوری و ایجاد یک bottleneck performance
  - وجود تحریم برای استفاده از ابزارهای متعدد
  - (این مورد میتواند به مدیریت پروژه برگرد و جزو ریسک ها نباشد) غریبه بودن اعضای تیم با این حیطه
- برای این موارد رویکردی که ما داریم این است که از ابزار ها و کتابخانه هایی استفاده کنیم که هم جامعه بزرگی داشته باشند و هم تا حد امکان متن باز باشند تا در صورت متوقف شدن آن ابزار در صورت نیاز خودمان بتوانیم تغییرات مورد نیاز را روی آن ها بدهیم
- برای تحریم میتوان از ابزارهای مختلفی استفاده کرد که این موضوع را حل کرد که در بحث این داک نمی‌گنجد درواقع مشکل اصلی این مورد بروز کندی و پیچیدگی در فرآیند develop/deploy است و به اصطلاح پروژه را کامل متوقف نمی‌کند.

## ۳.۷ ریسک های امنیتی

با توجه به اینکه این برنامه نیاز است کد ناشناس را بتواند اجرا کند، بحث ریسک های امنیتی اهمیت ویژه ای دارد

- امکان حمله های DDoS و پایین آمدن سیستم
  - امکان فرستادن کد های مخرب توسط کاربران به هدف
- برای حمله های DDoS چون فعلا برنامه ما در حالت دولوپ است، احتمال این جور حمله ها پایین تر است ولی در کل برای آینده میتوان از تکنیک های Mitigation DDoS مثلا استفاده از WAF و .... یا بهره گیری از خدمات provider cloud خود استفاده کرد
- برای کد های مخرب ما کارهای مختلفی برای ایزوله کردن محیط اجرای کد میتوانیم بکنیم مثلا قطع بودن اینترنت در محیط اجرا، اجرای کد ها در یک container جدا و ....

## ۴.۷ رقبای محصول

رقبا میتوانند به دو صورت مستقیم و غیر مستقیم برای ما مشکل درست کنند

- بهره گیری از ایده های ما و پیاده سازی فیچر های ما توسط خود
- از آنجا که سایر برنامه ها از ما بزرگ تر هستند، کاربران میتوانند ترجیح بدهند که از آن ها استفاده کنند
- وجود سوال های بیشتر و قوی تر در رقبا

## ۵.۷ بازاریابی

- عدم استفاده توسط کاربران
- این پروژه پیچیدگی های فنی ای دارد که ممکن است مخاطب های ما حوصله توضیحات ما درمورد آن را نداشته باشند و فکر کنند که برنامه ما یک فیچر را ندارد یا باگ دارد