



دانشگاه صنعتی نوشیروانی بابل
دانشکده برق و کامپیوتر

پایان نامه برای دریافت درجه‌ی کارشناسی
رشته‌ی کامپیوتر
سامانه اجرای کد از راه دور

دانشجو:

محمدعلی علی پناه

استاد راهنما:

دکتر غلامی

شهریور ۱۴۰۲

[این صفحه خالی است]

لَا إِلَهَ إِلَّا اللَّهُ مُحَمَّدٌ رَسُوْلُهُ

تقديم

این صفحه برای تقديم اثر از سوی دانشجو به افراد یا سازمان‌هایی که برای او مهم هستند در نظر گرفته شده است؛ در غیر این صورت، این صفحه را پاک کنید.

فهرست مطالب

۱	آشنایی با پروژه	۱
۳	مفاهیم پایه	۲
۳ Firecracker	۱-۲
۴ Fiber Go	۲-۲
۴ RabbitMQ	۳-۲
۴ PostgreSQL	۴-۲
۴ Next.js	۵-۲
۵	معماری پروژه	۳
۷ API	۱-۳
۸ VMVisor	۲-۳
۹ Frontline	۳-۳
۱۰ کلاینت	۴-۳

[این صفحه خالی است]

فهرست شکل‌ها

۳	۱-۲	نمایی از ساختار Firecracker	۳
۷	۱-۳	لیست API	۷
۸	۲-۳	وضعیت اجرا کد	۸
۱۰	۳-۳	طرح برخی از صفحه‌ها	۱۰
۱۱	۴-۳	لیست پروژه‌ها	۱۱

[این صفحه خالی است]

فصل ۱

آشنایی با پروژه

YARCEE مخفف Engine Execution Code Remote Another Yet سرویس است

[این صفحه خالی است]

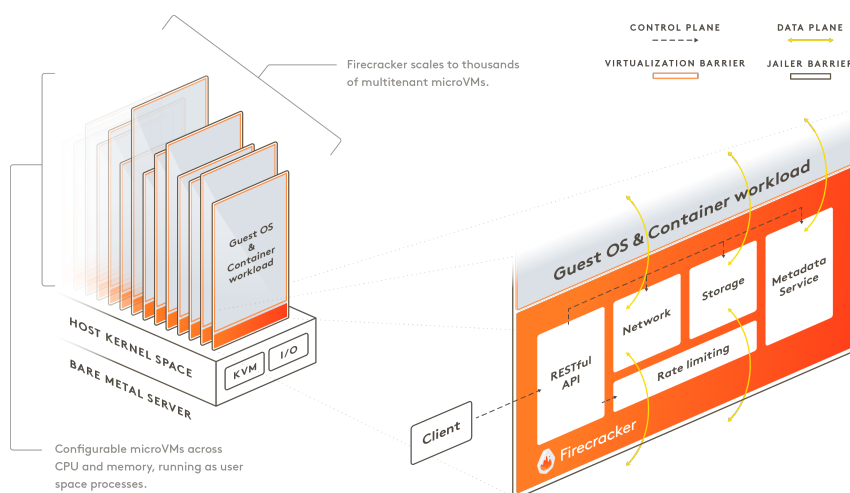
فصل ۲

مفاهیم پایه

در این فصل به معرفی تکنولوژی و فریمورک های به کار رفته در پروژه میپردازیم.

۱-۲ Firecracker

در قلب پروژه Firecracker قرار دارد. Firecracker یک مانیتور ماشین مجازی است که از KVM استفاده می کند و وظیفه اش ساخت و مدیریت ماشین های مجازی است. Firecracker توسط تیم وب سرویس آمازون توسعه داده شده و در پروژه Fargate و Lambda این شرکت نیز استفاده شده است.



شکل ۱-۲: نمایی از ساختار Firecracker

۲-۲ Fiber Go

زبان استفاده شده در میکروسرویس ها Go می باشد که توسط گوگل توسعه داده شده و از فریمورک Fiber استفاده شده که برای ساده تر شدن routing و middleware استفاده شده است. از دلایل استفاده از Go میتوان به سادگی و سرعت بالا اشاره کرد. همچنین این زبان در مسائل concurrency ابزارهای low-level زیادی در دسترس کاربر قرار می دهد.

۳-۲ RabbitMQ

RabbitMQ یک نرم افزار برای انتقال پیام بین سیستم ها است. در این پروژه درخواست اجرا کد در صف وارد می شود و توسط سرویسی پردازش می شود. دلیل استفاده از event-driven بلاک نشدن درخواست ها است. مزیت استفاده از RabbitMQ آسنکرون شدن سیستم است. پیام ها و وضعیت اجرای کد پشت یکدیگر بلاک نمی شوند. همچنین سیستم ها از وجود یکدیگر بی خبر هستند و وابستگی شان بهم کم می شود. به این معماری loosely-coupled می گویند.

۴-۲ PostgreSQL

دیتابیس اصلی استفاده شده PostgreSQL می باشد که از نوع رابطه ای است. جداول این نرم افزار شامل کاربران و کدها می باشد. دلیل انتخاب PostgreSQL مورد اطمینان بودن و سادگی این پایگاه داده بوده است. برای ارتباط و کوئری زدن از کتابخانه gorm زبان Go استفاده شده که ORM محبوبی است.

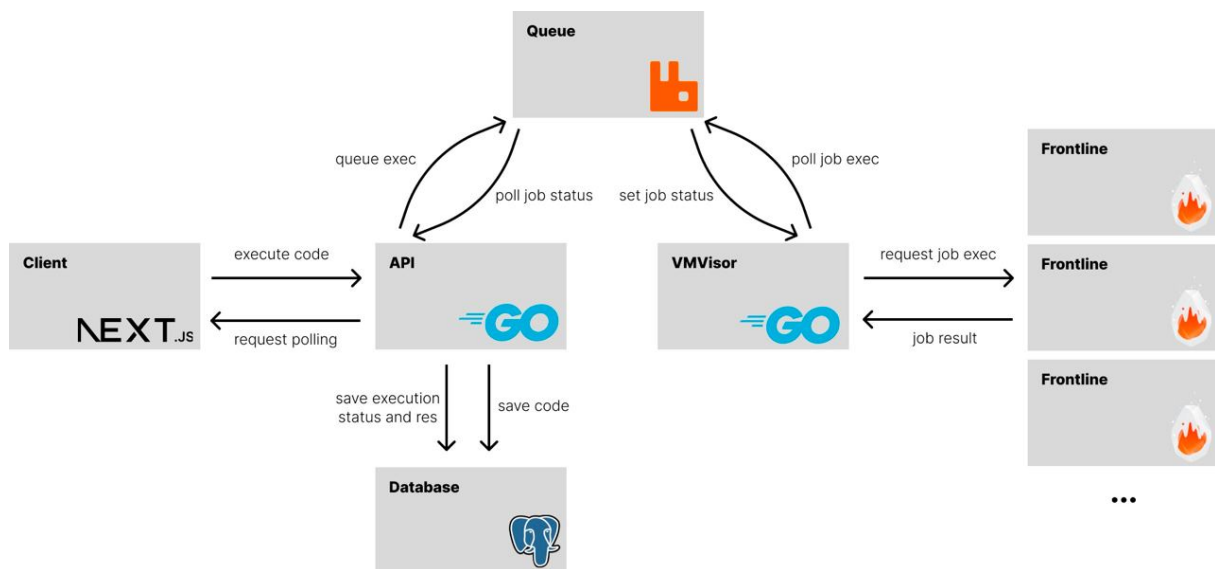
۵-۲ Next.js

فریمورک استفاده شده سمت کلاینت Next.js می باشد که از کتابخانه React برای رندر روی مرورگر استفاده میکند. دلیل استفاده از React ساده کردن پیاده سازی رابط کاربری توسط هوک ها و کامپوننت محور بودن آن است. زبان برنامه نویسی سمت کلاینت TypeScript است که به واسطه کامپایلر تبدیل به JavaScript می شود. دلیل استفاده از TypeScript اضافه شدن شی گرایی و تایپ در زمان کامپایل است.

فصل ۳

معماری پروژه

همانطور که در؟؟ می بینید این پروژه از معماری میکروسرویس بهره می برد. در ادامه به معرفی سرویس ها می پردازیم.



شکل ۳-۱: معماری پروژه

جدول ۳-۱: لیست سرویس ها

سرویس ها	توضیحات
API	سرویس API REST می باشد که وظیفه صحبت با دیتابیس و پاسخ به کلاینت را دارد
VMvisor	مدیریت VM های ساخته و دریافت و تغییر وضعیت درخواست های اجرا روی صف
Frontline	درون هر VM در حال اجراست و توسط پروسه فرزند کامپایلر زبان را صدا می زند
Client	کلاینت وظیفه نمایش رابط کاربری و ادیتور را دارد

۱-۳ API

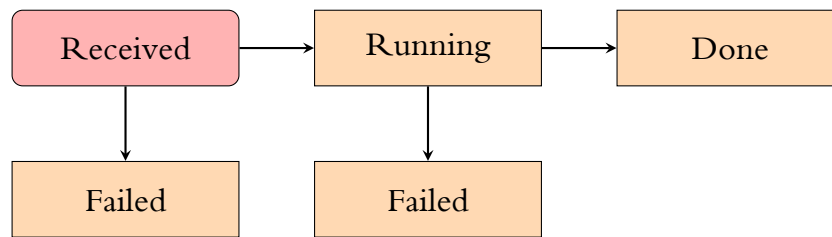
همانطور که اشاره کردم بخش API پروژه از زبان Go و فریمورک Fiber استفاده می‌کند. این پروژه لایه ورودی ما به بخش های داخلی سیستم می‌باشد.

در عکس زیر نمایی از api های موجود در پروژه مشاهده می‌شود. این api ها در چند دسته مختلف تقسیم بندی شده اند. sandbox برای ساخت یک پروژه جدید و اجرا آن می‌باشد. بخش auth برای ثبت نام و ورود کاربر است. بخش user برای دریافت اطلاعات کاربر وارد شده است. و health برای بررسی liveness و readiness سیستم در نظر گرفته شده است. وجود این مسیر باعث می‌شود در سیستم های مدیریت کانتینر مانند kubernetes از آمادگی سرویس اطمینان حاصل کرد.

Sandbox			^
GET	/api/sandbox/	Get user sandboxes	✓
POST	/api/sandbox/	Create sandbox	✓
DELETE	/api/sandbox/	Delete sandbox	✓
GET	/api/sandbox/{id}	Get sandbox	✓
PUT	/api/sandbox/{id}	Update sandbox	✓
POST	/api/sandbox/{id}/execute	Update sandbox	✓
Auth			^
POST	/api/sign-in/	Sign in	✓
POST	/api/sign-up/	Sign up	✓
User			^
GET	/api/user/	User auth details	✓
Health check			^
GET	/health_check/	Health check	✓

شکل ۳-۲: لیست API

شکل ۳-۳: وضعیت اجرا کد



VMVisor ۲-۳

۳-۳ Frontline

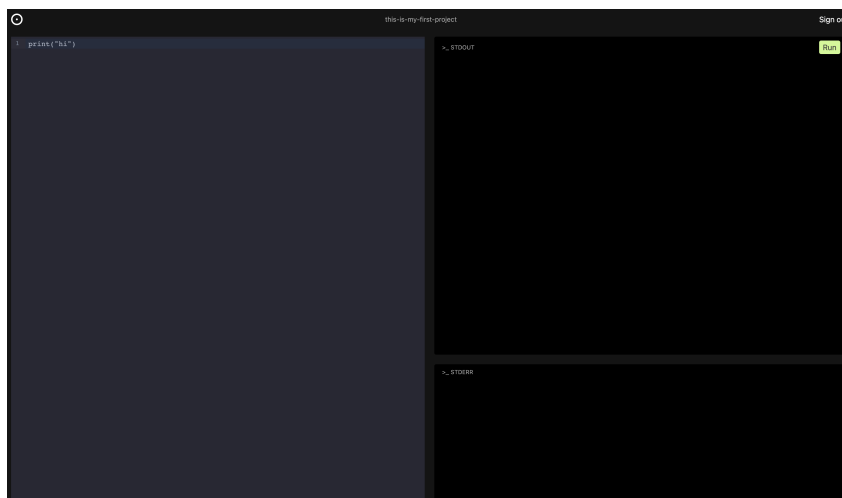
هـب

۴-۳ کلاینت

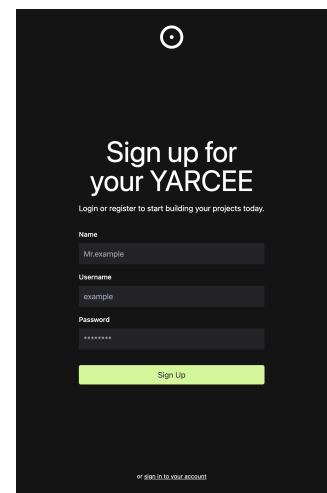
همانطور که اشاره کردیم کلاینت این پروژه از Next.js استفاده می‌کند. طراحی رابط کاربری در محیط Figma انجام شده است.

مراحل پیاده سازی به شرح زیر است:

۱. دیزاین توکن ها را استخراج و به پروژه اضافه می‌کنیم. مانند رنگ ها، فاصله ها و سایه ها
۲. المنت های دیزاین سیستم رو پیاده سازی می‌کنیم. کامپوننت هایی مانند دکمه، اینپوت و کانتینر
۳. با کنار هم قرار دادن المنت های دیزاین سیستم و کامپوننت های مخصوص هر بخش، صفحه را تکمیل می‌کنیم
۴. مراحل دریافت یا فرستان اطلاعات را انجام می‌دهیم
۵. مرحله ۳ و ۴ را برای هر صفحه تکرار میکنیم



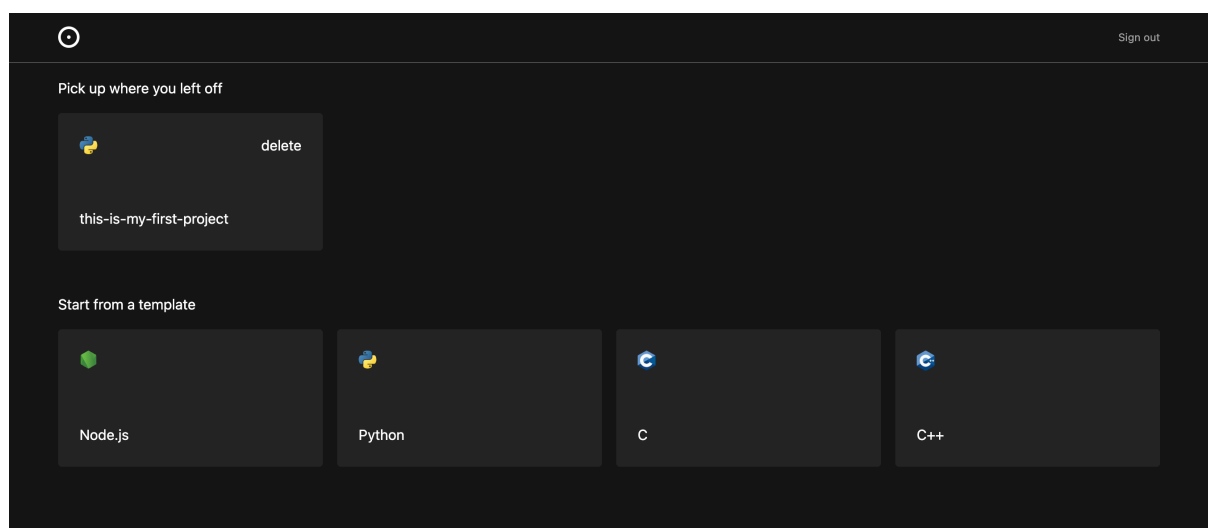
(ب) صفحه ادیت کد



(الف) صفحه ثبت نام

شکل ۴-۳: طرح برخی از صفحه ها

در شکل بالا می‌توانید نمونه ای از صفحه های پیاده سازی شده در این پروژه را مشاهده کنید. پس ثبت نام و ورود به سایت شما با صفحه داشبورد مواجه می‌شوید. در این صفحه می‌توانید به ادامه ویرایش پروژه قبلی خود بپردازید یا توسط قالب های از پیش تعیین شده پروژه جدیدی شروع کنید. اضافه کردن اکثر زبان های برنامه نویسی ممکن است ولی در حال حاضر از Python، Node.js، C++ و پشتیبانی می‌شود. اضافه کردن زبان جدید به سادگی امکان پذیر است که در ادامه در بخش Frontline به آن می‌پردازیم.



شکل ۳-۵: لیست پروژه ها

[این صفحه خالی است]

Abstract

YARCEE(Yet Another Remote Code Execution Engine) is a code running service that relies on Firecracker to spawn microVMs and execute the code over HTTP.

Keyword: Remote code execution - MicroVM - VMM



Babol Noshirvani University of Technology
Faculty (or School) of ...

A Thesis

*Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in
Computer*

Remote code execution engine

by:

Mohammad ali Ali panah

Supervisor:

Dr.Gholami

August, 2023