ءانب توبور يصصق موعدم ءاكذلاب يعانطصلاا

تمت كابتة هذه كتابة مدونة المدمج بمساعدة [illegible]4.

جدول المحتويات

- مقدمة
- هندسة مشروع الشبكة
  - الخلفية الواجهة [illegible]
    - إعداد تطبيق [illegible]
    - المراقبة التسجيل
    - الطلبات مع الجاذبية
  - الأمامية الواجهة [illegible]
    - مكونات [illegible]
    - تكامل [illegible]
- النشر
  - نشر الصندوق
  - بيئة [illegible]
  - بيئة [illegible]
  - بيئة [illegible]
- بيئة [illegible] وشهادة [illegible] من [illegible] [illegible]
  - تحديد خطوات العمل مع الأصول المسموح بها
  - إعادة توجيه [illegible] إلى [illegible]
  - بيئة الموقع الرئيسي لـ example.com
  - بيئة [illegible] لـ api.example.com
- الخاتمة

مقدمة

الاصطناعي. بالذكاء المدعوم القصص روبوت تطبيق وتنفيذ بناء حول شاملًا دليلًا المدونة هذه توفر و[illegible] و[illegible] [illegible] نستخدم ويب. واجهة باستخدام مخصصة قصص إنشاء المشروع يتضمن

و[illegible] لل[illegible] د[illegible] نستخدم ذلك، إلى بالإضافة .[illegible] على بالنشر وتقوم للتطوير
[illegible] و[illegible] مع و[illegible]، [illegible] خلال من [illegible] إدارة يتم السجلات. لإدارة و[illegible] و[illegible]
الطلبات. ورؤوس [illegible] شهادات لإدارة كبوابة [illegible]

المشروع هندسة

الخلفية الواجهة [illegible] تم بناء الجزء الخلفي [illegible] للمشروع [illegible]، إطار وهو [illegible] باستخدام
طلبات مع الجزء الخلفي [illegible] على يتولى .[illegible] لغة في [illegible] على يعتمد الويب لتطبيقات الوزن خفيف عمل
المراقبة. لأغراض [illegible] مع والتكامل التطبيق، أنشطة وتسجيل البيانات، قاعدة وإدارة، [illegible]

الخلفية: الواجهة لمكونات تفصيلًا إليك

1. إعداد تطبيق [illegible]:
   - يتم تهيئة تطبيق [illegible] من العديد لاستخدام وتكوينه [illegible] لمثل الإضافات
     [illegible] و[illegible] [illegible] [illegible] [illegible] [illegible] المصادر عبر الموارد مشاركة مع
     البيانات. قواعد تحرير لإدارة
   - يتم تهيئة تطبيق التطبيقات وتمكين [illegible] حاسم للسلامة [illegible] المصادر. عبر
   - يتم تهيئة قاعدة البيانات مع إعدادات وإعداد افتراضية، السجلات إدخال لتسنيق مخصص سجل
     [illegible] لـ.

```
from flask import Flask
from flask_cors import CORS
from .routes import initialize_routes
from .models import db, insert_default_config
from flask_migrate import Migrate
import logging
from logging.handlers import RotatingFileHandler
from prometheus_client import Counter, generate_latest, Gauge

app = Flask(__name__)
app.config.from_object('api.config.BaseConfig')

db.init_app(app)
initialize_routes(app)
CORS(app)
migrate = Migrate(app, db)
```

2. والمراقبة: الأحداث تسجيل

- مُنسق باستخدام وتنسيقها السجلات لإدارة ملفات RotatingFileHandler التطبيق يستخدم مخصص.
- تم جمع مقاييس [illegible] في التطبيق لتتبع عدد الطلبات وزمن الاستجابة.

```
REQUEST_COUNT = Counter('flask_app_request_count', '               Flask', ['method', 'endpoint', 'http
REQUEST_LATENCY = Gauge('flask_app_request_latency_seconds', '          ', ['method', 'endpoint'])

def setup_loggers():
    logstash_handler = RotatingFileHandler('app.log', maxBytes=100000000, backupCount=1)
    logstash_handler.setLevel(logging.DEBUG)
    logstash_formatter = CustomLogstashFormatter()
    logstash_handler.setFormatter(logstash_formatter)

root_logger = logging.getLogger()
root_logger.setLevel(logging.DEBUG)
root_logger.addHandler(logstash_handler)
```

تم تعيين root_logger للحصول على السجل الأساسي باستخدام logging.getLogger(). ثم تم تعيين مستوى التسجيل إلى DEBUG باستخدام setLevel(logging.DEBUG). أخيرًا، تمت إضافة معالج السجل الأساسي إلى logstash_handler باستخدام addHandler(logstash_handler).

```
app.logger.addHandler(logstash_handler)
werkzeug_logger = logging.getLogger('werkzeug')
werkzeug_logger.setLevel(logging.DEBUG)
werkzeug_logger.addHandler(logstash_handler)

setup_loggers()
```

3. معالجة الطلبات:

- [illegible] تتبع فريد إنشاء طلب، لكل وقبل السجل المقاييس تحديث التطبيق يقوم الطلب. تدفق لمراقبة

```
def generate_trace_id(length=4):
    characters = string.ascii_letters + string.digits
    return ''.join(random.choice(characters) for _ in range(length))

@app.before_request
def before_request():
```

```
    request.start_time = time.time()
    trace_id = request.headers.get('X-Trace-Id', generate_trace_id())
    g.trace_id = trace_id
```

العربية: إلى الكود ترجمة

```
@app.before_request
def before_request():
    request.start_time = time.time()
    trace_id = request.headers.get('X-Trace-Id', generate_trace_id())
    g.trace_id = trace_id
```

في هذا الكود، يتم تنفيذ الدالة before_request قبل كل طلب يتم إرساله إلى التطبيق. يتم تسجيل وقت بدء الطلب باستخدام request.start_time = time.time(). ثم يتم استخراج trace_id من رأس الطلب request.headers.get('X-Trace-Id')، وإذا لم يتم العثور على trace_id في الرأس، يتم إنشاؤه باستخدام الدالة generate_trace_id(). أخيرًا، يتم تخزين trace_id في الكائن g الذي يمكن الوصول إليه في جميع أنحاء التطبيق.

```
@app.after_request
def after_request(response):
    response.headers['X-Trace-Id'] = g.trace_id
    request_latency = time.time() - getattr(request, 'start_time', time.time())
    REQUEST_COUNT.labels(method=request.method, endpoint=request.path, http_status=response.status_code
    REQUEST_LATENCY.labels(method=request.method, endpoint=request.path).set(request_latency)
    return response
```

الأمامية الواجهة تم بناء واجهة المشروع باستخدام [illegible]، وهي مكتبة [illegible] لبناء واجهة وتوفر القص نصوص لإدارة الخلفية [illegible] التطبيقات برمجة واجهة مع تتفاعل المستخدم. واجهات المخصصة. القص وإدارة لإنشاء تفاعلية مستخدم

1. مكونات [illegible]:

   [illegible] المكون الرئيسي المتعامل مع الإدخال مستخدم الطلبات لـ القص التفاعلي مع واجهة برمجة القص. هذه لإدارة الخلفية [illegible] التطبيقات

```
import React, { useState, useEffect } from 'react';
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
```

```
import { apiFetch } from './api';
import './App.css';

function App() {
  const [prompts, setPrompts] = useState([]);
  const [newPrompt, setNewPrompt] = useState('');
  const [isLoading, setIsLoading] = useState(false);

useEffect(() => {
  fetchPrompts();
}, []);

const fetchPrompts = async () => {
  setIsLoading(true);
  try {
    const response = await apiFetch('prompts');
    if (response.ok) {
      const data = await response.json();
      setPrompts(data);
    } else {
      toast.error('            '(;
    }
  } catch (error) {
    toast.error('                '(;
  } finally {
    setIsLoading(false);
  }
};

const addPrompt = async () => {
  if (!newPrompt) {
    toast.warn('      Prompt              '(;
    return;
  }
  setIsLoading(true);
  try {
    const response = await apiFetch('prompts', {
```

```
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ content: newPrompt }),
    });
    if (response.ok) {
      fetchPrompts();
      setNewPrompt('');
      toast.success('         Prompt    '(;
    } else {
      toast.error('           Prompt');
    }
  } catch (error) {
    toast.error('               Prompt');
  } finally {
    setIsLoading(false);
  }
};

const deletePrompt = async (promptId) => {
  setIsLoading(true);
  try {
    const response = await apiFetch(`prompts/${promptId}`, {
      method: 'DELETE',
    });
    if (response.ok) {
      fetchPrompts();
      toast.success('       Prompt    '(;
    } else {
      toast.error('          Prompt');
    }
  } catch (error) {
    toast.error('                  Prompt');
  } finally {
    setIsLoading(false);
  }
```

```
};

      return (
        <div className="app">
          <h1>                      >/h1>
          <div>
            <input
              type="text"
              value={newPrompt}
              onChange={(e) => setNewPrompt(e.target.value)}
              placeholder=   " "
            </
            <button onClick={addPrompt} disabled={isLoading}>       >/button>
          </div>
          {isLoading ? (
            <p>       >.../p>
          ) : (
            <ul>
              {prompts.map((prompt) => (
                <li key={prompt.id}>
                  {prompt.content}
                  <button onClick={() => deletePrompt(prompt.id)}>  >/button>
                </li>
              ))}
            </ul>
          )}
          <ToastContainer />
        </div>
      );
    }

    export default App;
    ```

2.   API:
    -                          (API)   fetch   .
```
```

```javascript
export const apiFetch = (endpoint, options) => {
  return fetch(`https://api.yourdomain.com/${endpoint}`, options);
};
```

رشن ال

تم رشن عورشمال ىلع □□□، عم إدارة □□□ متت من لال خ □□□□□□□□. □□□□□□□□□□□□و متي دام خ تسا □□□□□□ و□□ و□□□□□□□□□□□□□□□□□□ لدصرلل □□□□□□□□□□□□ نس خ تدم الطلبات. وتيرو سا □□□ ش هادا ت لإدارة كبابوا ة □□□ ال سجلات. ل إدارة و□□□□□□□□□ □□□□□□

1. تبي كس: رشن ال

   □ مدخ تسن □□□□□□□ تة تأ ل مام رشن ال لث م داد إع دات لجم ال ية لحم ال، دة ي عب وال نة زام ومل ات، ف ل م ال
   ال أذون ات. وض بط

```
from fabric import task
from fabric import Connection

server_dir = '/home/project/server'
web_tmp_dir = '/home/project/server/tmp'

@task
def prepare_remote_dirs(c):
    if not c.run(f'test -d {server_dir}', warn=True).ok:
        c.sudo(f'mkdir -p {server_dir}')
        c.sudo(f'chmod -R 755 {server_dir}')
        c.sudo(f'chmod -R 777 {web_tmp_dir}')
        c.sudo(f'chown -R ec2-user:ec2-user {server_dir}')
```

ت رجمة:

```
@task
def prepare_remote_dirs(c):
    if not c.run(f'test -d {server_dir}', warn=True).ok:
        c.sudo(f'mkdir -p {server_dir}')
        c.sudo(f'chmod -R 755 {server_dir}')
        c.sudo(f'chmod -R 777 {web_tmp_dir}')
        c.sudo(f'chown -R ec2-user:ec2-user {server_dir}')
```

:ةظحلام تم ظافحلا ىلع دوكلا اذه وأ لوحتي رماوأ ءاسمأو تارييغت ،ةيزيلجنإلا بايتلا متي لا
.قايس يف ةدعاً ةجرمبرلا ةمجرتملا

```
@task
def deploy(c, install='false'):
    prepare_remote_dirs(c)
    pem_file = './aws-keypair.pem'
    rsync_command = (f'rsync -avz --exclude="api/db.sqlite3" '
                     f'-e "ssh -i {pem_file}" --rsync-path="sudo rsync" '
                     f'{tmp_dir}/ {c.user}@{c.host}:{server_dir}')
    c.local(rsync_command)
    c.sudo(f'chown -R ec2-user:ec2-user {server_dir}')
```

.يه اذكما ةيزيلجنإلا ءاسمأ ىلع ظافحلا عم ةيبرعلا ةغللا ىلإ أعلاه دوكلا ةمجرت تمت

2. ةئيهت [illegible]:

   - نمضتي دادعإ [illegible] تادادعإل ةلثمأ [illegible]، ةدقعلا [illegible]، والشبكة.

```
cluster.name: my-application
node.name: node-1
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 0.0.0.0
http.port: 9200
discovery.seed_hats: ["127.0.0.1"]
cluster.initial_master_nodes: ["node-1"]
```

3. نيوكت [illegible]:

   - نمضتي دادعإ [illegible] تانيوكت مداخل [illegible] تافيضمو [illegible].

```
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.hosts: ["http://localhost:9200"]
```

4. نيوكت [illegible]:

- تم تكوين □□□□□□□□ لقراءة ملفات السجلات، وتحليلها، وإخراج السجلات الجديدة إلى □□□□□ □□□□□□□□□.

```
input {
  file {
    path => "/home/project/server/app.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
  }
}

filter {
  json {
    source => "message"
  }
}
```

تمت ترجمة الكود أعلاه إلى:

```
  }
  json {
      => "    "
  {
}

output {       elasticsearch {         hosts => ["http://localhost:9200"]         index
=> "flask-logs-%{+YYYY.MM.dd}"       }     }
```

تكوين □□□□□ وشهادة □□□ من □□□□□ □□□□□□□

تأكد من تحديث مدخل الآمن، الاتصال لضمان □□□□□ عكسي ولكي □□□□□ و□□□□□□□ للوصول على شهادات □□□. قم بتكوين ما يلي □□□□□ لإعادة توجيه □□□□ إلى □□□□□ وإعداد شهادات □□□.

```
server {
    listen 80;
    server_name example.com;
```

```
    #               HTTP    HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name example.com;

    #        SSL
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;

    #      SSL
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    #
    location / {
        proxy_pass http://localhost:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

في هذا التكوين: تم إعادة توجيه جميع طلبات HTTP إلى HTTPS باستخدام return 301. تم تعيين شهادة SSL باستخدام ssl_certificate و ssl_certificate_key. تم إعدادات استخدام أحدث البروتوكولات الآمنة. تم تكوين الخادم لتمرير الطلبات إلى خادم Node.js يعمل على المنفذ 3000 باستخدام proxy_pass.

1. قم بتعطيل الخادم الافتراضي لـ Nginx: هذا ضروري لتجنب أي تعارض مع تكوينك المخصص.

```
map $http_origin $cors_origin {
    default "https://example.com";
    "http://localhost:3000" "http://localhost:3000";
    "https://example.com" "https://example.com";
```

```
        "https://www.example.com" "https://www.example.com";
    }
```

2. إعادة توجيه □□□□ إلى □□□□□:

```
    server {
        listen 80;
        server_name example.com api.example.com;

        return 301 https://$host$request_uri;
    }
    ```
```

3. `example.com`:

```
    ```nginx
    server {
        listen 443 ssl;
        server_name example.com;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;

        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;
        ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";

        root /home/project/web;
        index index.html index.htm index.php default.html default.htm default.php;

        location / {
            try_files $uri $uri/ =404;
        }
```

إلى: أعلاه الكود ترجمة تمت

```
        location / {
            try_files $uri $uri/ =404;
        }
```

ب- الخاصة التكوين بلغة مكتوب أنه حيث ترجمة إلى يحتاج ولا، [illegible] لخادم تكوين هو أعلاه الكود ملاحظة: [illegible].

```
        location ~ .*\.(gif|jpg|jpeg|png|bmp|swf)$ {
            expires 30d;
        }
```

إلى: أعلاه الكود ترجمة تمت

```
        location ~ .*\.(gif|jpg|jpeg|png|bmp|swf)$ {
            expires 30d;
        }
```

، [illegible]، [illegible]، [illegible]، [illegible]، [illegible] مثل والفيديو الصور لملفات [illegible] موقع تحديد يتم الكود، هذا في يومًا. 30 لمدة [illegible] المؤقت التخزين صلاحية وتعيين [illegible]

```
        location ~ .*\.(js|css)?$ {
            expires 12h;
        }
```

إلى: أعلاه الكود ترجمة تمت

```
        location ~ .*\.(js|css)?$ {
            expires 12h;
        }
```

يتم التي و[illegible] ملفات لجميع ساعة 12 لمدة [illegible] صلاحية تعيين يتم الكود، هذا في 12 لمدة [illegible] المؤقت التخزين ذاكرة في الملفات هذه سيخزن المتصفح أن يعني هذا الخادم. من طلبها الخادم. من أخرى مرة طلبها إلى يحتاج أن قبل ساعة

```
        error_page 404 /index.html;
    }
    ```
```

4. API `api.example.com`:

```
    ```nginx
    server {
```

```
        listen 443 ssl;
        server_name api.example.com;
    ```

        ssl_certificate /etc/letsencrypt/live/example.com-0001/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/example.com-0001/privkey.pem;

```nginx
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;
        ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";

        location / {
            #         Access-Control
            more_clear_headers 'Access-Control-Allow-Origin';

#            CORS      (preflight)
if ($request_method = 'OPTIONS') {
    add_header 'Access-Control-Allow-Origin' $cors_origin;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
    add_header 'Access-Control-Allow-Headers' 'Origin, Content-Type, Accept, Authorization, X-Client-In
    add_header 'Access-Control-Max-Age' 3600;
    return 204;
}

add_header 'Access-Control-Allow-Origin' $cors_origin always;
add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE' always;
add_header 'Access-Control-Allow-Headers' 'Origin, Content-Type, Accept, Authorization, X-Client-Info, X

nginx             proxy_pass http://127.0.0.1:5000/;             proxy_set_header Host
$host;             proxy_set_header X-Real-IP $remote_addr;             proxy_set_header
X-Forwarded-For $proxy_add_x_forwarded_for;             proxy_set_header X-Forwarded-Proto
$scheme;             proxy_connect_timeout 600s;             proxy_send_timeout 600s;
proxy_read_timeout 600s;             send_timeout 600s;         }     }
```

الخلاصة

وأدوات ممارسات مستخدمًا الاصطناعي، بالذكاء مدعوم قصصي بوت لتطبيق قوية بنية المشروع هذا يعرض والتكامل للطلبات فعالة معالجة يضمن مما ،[illegible] باستخدام الخلفية الواجهة بناء تم حديثة. ويب تطوير مستخدم واجهة ،[illegible] باستخدام المبنية الأمامية، الواجهة توفر والمراقبة. للتسجيل مختلفة خدمات مع [illegible] ومجموعة الآمن، للاتصال و[illegible] للنشر، [illegible] من الاستفادة خلال من القصص. محفزات لإدارة تفاعلية الجمع قوة الشامل الإعداد هذا يوضح للصيانة. والقابلية والموثوقية التوسع قابلية نضمن السجلات، لإدارة سلسة. مستخدم تجربة لتقديم الحديثة التقنيات بين