

- A. CSS Test
- B. API Test

CSS and API Tests can be seen on the Vue app

### C. SQL Test

-- Result 1

```
SELECT string_field_0 FROM `portfolio-397204.BTI.asset` GROUP BY string_field_0
```

-- Result 2

```
WITH temp AS (
  SELECT string_field_1, string_field_0
  FROM `portfolio-397204.BTI.asset`
)
```

```
SELECT string_field_1, STRING_AGG(string_field_0, ', ') AS asset
FROM temp
```

\*string\_field\_0 is item and string\_field\_1 is user\_id  
this query tested on BigQuery

```

1  --Result 1
2  SELECT string_field_0 FROM `portfolio-397204.BTI.asset` GROUP BY string_field_0
3
4
5  --Result 2
6  WITH temp AS (
7    SELECT string_field_1, string_field_0
8    FROM `portfolio-397204.BTI.asset`
9  )
10
11 SELECT string_field_1, STRING_AGG(string_field_0, ', ') AS asset
12 FROM temp

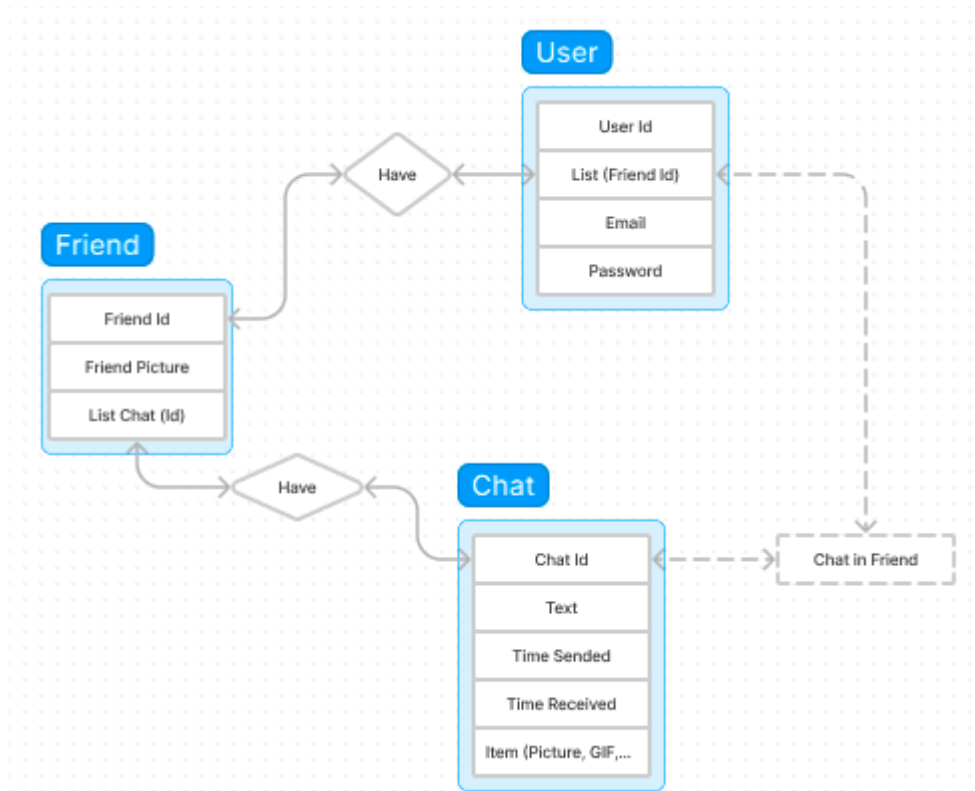
```

Press Alt+F1 for Accessibility Options.

**Query results** 
[SAVE RESULTS](#)
[EXPLORE DATA](#)

JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	string_field_1	asset				
1	wildan	notebook, bag				
2	zaki	notebook, bag, mobile phone				

#### D. ERD Test



#### E. Logical Test

in E. Logical Test Folder

#### F. Jelaskan istilah dibawah dengan bahasa sendiri

##### 1. Cypress

Cypress adalah alat testing penerimaan yang dapat membantu melakukan testing aplikasi web dan API secara otomatis. Cypress menggunakan JavaScript untuk menjalankan tesnya, sehingga kita dapat menggunakan pengetahuan JavaScript yang kita miliki untuk menulis tes.

##### 2. Kubernetes

Kubernetes adalah open source cluster system management yang dapat digunakan untuk meningkatkan skalabilitas dan mengelola aplikasi yang kompleks. Kubernetes menggunakan konsep kontainer untuk mengemas aplikasi, sehingga dapat dengan mudah dipindahkan dan diskalakan.

##### 3. ORM

ORM adalah singkatan dari Object Relational Mapping. ORM adalah alat yang membantu memanipulasi data di database relasional menggunakan objek. ORM dapat membuat kode lebih mudah dibaca dan dipelihara.

##### 4. Class

Class adalah sebuah blueprint untuk membuat objek. Class mendefinisikan properti dan metode yang dimiliki oleh objek. Objek adalah instansiasi dari class. Class tersedia di beberapa bahasa seperti Java, C++, JS/TS, Kotlin, dan bahasa pemrograman lainnya.

##### 5. Function

Function adalah blok kode yang dapat diulang. Function dapat digunakan untuk mengelompokkan kode yang terkait dan membuatnya lebih mudah dibaca dan dipelihara.

## 6. DevOps

DevOps adalah singkatan dari Development Operations. DevOps adalah pendekatan yang menyatukan pengembangan dan operasi untuk mempercepat proses pengiriman (deployment) aplikasi. DevOps menggunakan berbagai macam alat dan praktik untuk membuat proses pengiriman aplikasi lebih efisien dan efektif.

## 7. Docker

Docker adalah alat yang memungkinkan untuk mengemas aplikasi ke dalam kontainer. Kontainer adalah paket perangkat lunak yang berisi semua hal yang dibutuhkan aplikasi untuk berjalan, termasuk kode, pustaka, dan sistem operasi. Docker dapat digunakan untuk menskalakan dan mengelola aplikasi secara lebih efisien. Kontainer ini bisa diorkestrasi menggunakan tools cluster system management sebagai contohnya Kubernetes.

G. Berikan 3 Architecture Application di era sekarang, tentukan teknologi yang menurut Anda terbaik, dan berikan alasannya?

Saya merasa ada banyak arsitektur di dunia ini dan tidak ada arsitektur yang sempurna. Saya sendiri merasa penggunaan arsitektur harus disesuaikan dengan kebutuhan atau use case masing-masing masalah. Namun jika saya boleh memberikan 3 rekomendasi arsitektur yang sering saya pakai dan lumayan saya sukai, 3 arsitektur tersebut adalah:

### 1. Microservice

Arsitektur microservice adalah arsitektur yang membagi sistem menjadi beberapa bagian kecil yang independen, yang disebut microservices. Masing-masing microservice bertanggung jawab untuk satu fungsi tertentu.

Poin-poin positif:

- Scalability: Arsitektur microservice dapat diskala dengan mudah dengan menambahkan atau mengurangi microservices sesuai kebutuhan.
- Resilience: Arsitektur microservice lebih resilient terhadap kegagalan karena setiap microservice dapat diandalkan secara mandiri.
- Agility: Arsitektur microservice lebih agile karena memungkinkan tim pengembangan untuk bekerja secara independen pada microservices yang berbeda.

Poin-poin negatif:

- Complexity: Arsitektur microservice dapat menjadi kompleks untuk dikelola, terutama ketika sistem menjadi besar.
- Communication: Komunikasi antar microservices dapat menjadi tantangan, terutama ketika microservices menggunakan teknologi yang berbeda.
- Security: Arsitektur microservice dapat meningkatkan kerentanan keamanan karena setiap microservice memiliki API yang terbuka.

### 2. Single Page App (SPA)

Arsitektur SPA adalah arsitektur aplikasi web yang hanya memuat satu halaman HTML ke dalam browser pengguna. Halaman ini kemudian diperbarui secara dinamis oleh JavaScript.

Poin-poin positif:

- Performance: SPA dapat memberikan pengalaman pengguna yang lebih lancar karena tidak perlu memuat ulang halaman secara keseluruhan.
- UX: SPA dapat memberikan pengalaman pengguna yang lebih baik karena memungkinkan pengembang untuk mengontrol sepenuhnya bagaimana halaman ditampilkan dan berinteraksi dengan pengguna.

Poin-poin negatif:

- SEO: SPA dapat menurunkan peringkat SEO aplikasi web karena kontennya tidak dapat diindeks oleh mesin pencari.
- Development: SPA dapat lebih sulit dikembangkan daripada aplikasi web tradisional karena membutuhkan pengetahuan JavaScript yang lebih mendalam.

### 3. Progressive Web App (PWA)

Arsitektur PWA adalah arsitektur aplikasi web yang menggabungkan keunggulan web dan aplikasi native. PWA dapat diinstal di perangkat pengguna dan berjalan offline.

Poin-poin positif:

- Performance: PWA dapat memberikan pengalaman pengguna yang lebih lancar karena dapat dimuat secara offline dan berjalan di latar belakang.
- UX: PWA dapat memberikan pengalaman pengguna yang lebih baik karena dapat memanfaatkan fitur-fitur perangkat seperti notifikasi dan push.
- SEO: PWA dapat meningkatkan peringkat SEO aplikasi web karena kontennya dapat diindeks oleh mesin pencari.

Poin-poin negatif:

- Development: PWA dapat lebih sulit dikembangkan daripada aplikasi web tradisional karena membutuhkan pengetahuan tentang teknologi web dan native.

Honorable Mention:

#### 1. Serverless

Arsitektur serverless adalah arsitektur yang menggunakan layanan cloud untuk menjalankan kode tanpa perlu mengelola infrastruktur server.

Poin-poin positif:

- Scalability: Arsitektur serverless dapat diskala dengan mudah dengan menambahkan atau mengurangi sumber daya cloud sesuai kebutuhan.
- Cost-effectiveness: Arsitektur serverless dapat lebih hemat biaya daripada arsitektur tradisional karena hanya perlu membayar untuk sumber daya yang digunakan.
- Agility: Arsitektur serverless lebih agile karena memungkinkan tim pengembangan untuk fokus pada pengembangan aplikasi tanpa perlu khawatir tentang infrastruktur server.

Poin-poin negatif:

- Complexity: Arsitektur serverless dapat menjadi kompleks untuk dikelola, terutama ketika aplikasi menjadi besar.

- Vendor lock-in: Arsitektur serverless dapat membuat aplikasi bergantung pada vendor cloud tertentu, yang dapat membatasi pilihan di masa depan.