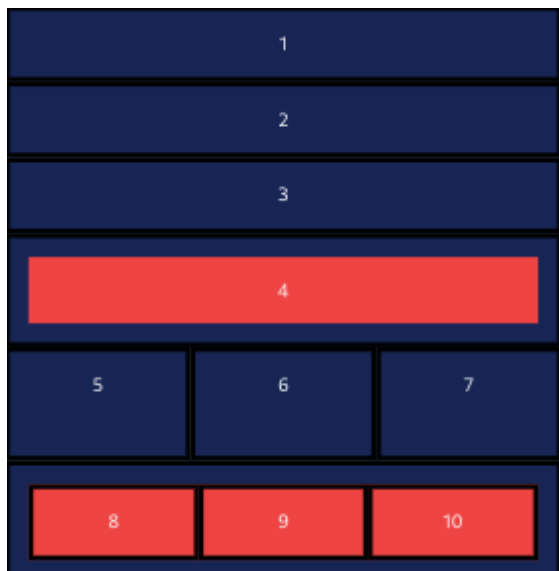
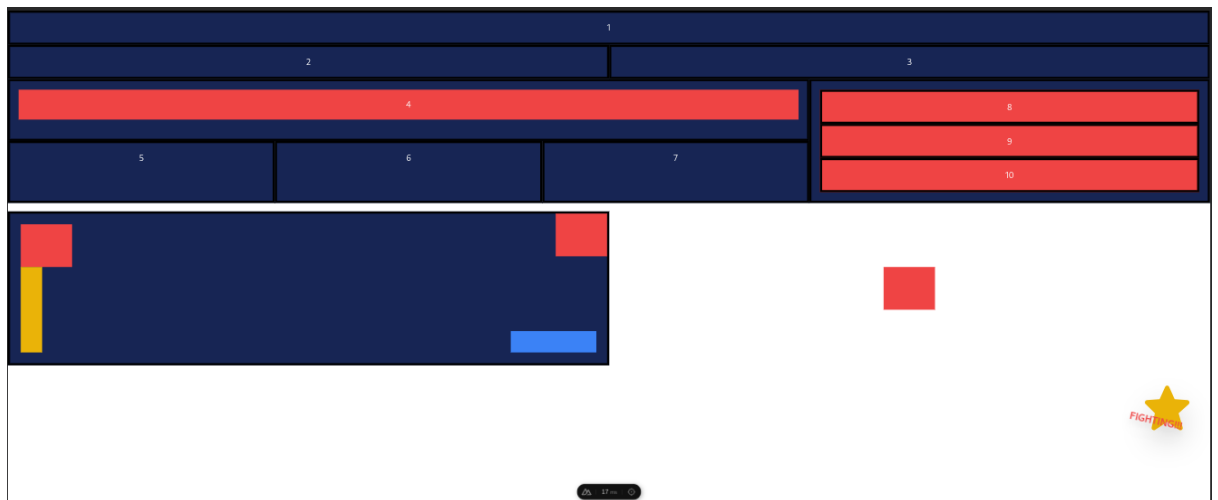


A.



B.

```
<template>
<div>
  <div class="grid grid-cols-3 gap-2 m-4">
    <button class="text-white p-4 rounded bg-indigo-500 shadow-md flex items-center
justify-center">
      <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 mr-2" fill="none"
viewBox="0 0 24 24"
stroke="currentColor">
        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M3 12l2-2m0 0l7-7 7 7M5 10v10a1 1 0 01 1h3m10-11l2 2m-2-2v10a1 1
0 01-1 1h-3m-6 0a1 1 0 001-1v-4a1 1 0 011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6" />
      </svg>
```

```

Marketing
</button>
<button class="p-4 rounded bg-white text-indigo-500 shadow-md flex items-center
justify-center">
  <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 mr-2" fill="none"
viewBox="0 0 24 24"
stroke="currentColor">
    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M5.121 17.804A13.937 13.937 0 0112 16c2.5 0 4.847.655 6.879
1.804M15 10a3 3 0 11-6 0 3 3 0 016 0zm6 2a9 9 0 11-18 0 9 9 0 0118 0z" />
  </svg>
  IT
</button>
<button class="p-4 rounded bg-white text-indigo-500 shadow-md flex items-center
justify-center">
  <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 mr-2" fill="none"
viewBox="0 0 24 24"
stroke="currentColor">
    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M17 8h2a2 2 0 012 2v6a2 2 0 01-2 2h-2v4l-4-4H9a1.994 1.994 0
01-1.414-.586m0 0L11 14h4a2 2 0 002-2V6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0 002
2h2v4l.586-.586z" />
  </svg>
  Design
</button>
</div>
<div class="shadow-xl border border-gray-100 font-light p-8 rounded text-gray-500
bg-white mt-6 mx-4">
  <h1 class="text-green-500">{{ jobPost.title }}</h1>
  <p><strong>Description:</strong></p>
  <div v-html="jobPost.description.txt"></div>
  <h5><strong>Skill :</strong></h5>
  <ul>
    <li v-for="skill in jobPost.ptJobSkills" :key="skill.id">
      {{ skill.name }}
    </li>
  </ul>
</div>
<div class="shadow-xl border border-gray-100 font-light p-8 rounded text-gray-500
bg-white mt-6 mx-4">
  <div class="grid grid-cols-2 gap-2 my-2 mx-4">
    <div class="mb-4">
      <label class="block text-gray-700 text-sm font-bold mb-2" for="displayName">
        Yourname
      </label>
      <input v-model="displayName"
class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

```

```

        id="displayName" type="text" placeholder="Username" />
    </div>
    <div class="mb-4">
        <label class="block text-gray-700 text-sm font-bold mb-2" for="phone">
            Mobile number
        </label>
        <input v-model="phone"
            class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="phone" type="text" placeholder="Mobile number" />
    </div>
</div>
<div class="grid grid-cols-2 gap-2 my-2 mx-4">
    <div class="mb-4">
        <label class="block text-gray-700 text-sm font-bold mb-2" for="email">
            Email
        </label>
        <input v-model="email"
            class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="email" type="text" placeholder="Email" />
    </div>
    <div class="mb-4">
        <label class="block text-gray-700 text-sm font-bold mb-2" for="message">
            Message
        </label>
        <input v-model="message"
            class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
            id="message" type="text" placeholder="Message" />
    </div>
</div>
<div class="container ">
    <div id="images-container"></div>
    <div class="flex my-4">
        <div id="multi-upload-button"
            class="inline-flex items-center px-4 py-2 bg-gray-600 border border-gray-600
rounded-l font-semibold cursor-pointer text-sm text-white tracking-widest hover:bg-gray-500
active:bg-gray-900 focus:outline-none focus:border-gray-900 focus:ring focus:ring-gray-300
disabled:opacity-25 transition ">
            Click to browse
        </div>
        <div class="w-4/12 lg:w-3/12 border border-gray-300 rounded-r-md flex
items-center justify-between">
            <span id="multi-upload-text" class="p-2"></span>
            <button id="multi-upload-delete" class="hidden"
                @click="removeMultiUpload">

```

```

        <svg xmlns="http://www.w3.org/2000/svg" class="fill-current text-red-700
w-3 h-3"
        viewBox="0 0 320 512">
        <path
            d="M310.6 361.4c12.5 12.5 12.5 32.75 0 45.25C304.4 412.9 296.2
416 288 416s-16.38-3.125-22.62-9.375L160 301.3L54.63 406.6C48.38 412.9 40.19 416 32
416S15.63 412.9 9.375 406.6c-12.5-12.5-12.5-32.75 0-45.25l105.4-105.4L9.375
150.6c-12.5-12.5-12.5-32.75 0-45.25s32.75-12.5 45.25 0L160 210.8l105.4-105.4c12.5-12.5
32.75-12.5 45.25 0s12.5 32.75 0 45.25l-105.4 105.4L310.6 361.4z" />
        </svg>
        </button>
    </div>
</div>
<input type="file" id="multi-upload-input" class="hidden" ref="fileInput"
@change="handleFileChange" />
</div>
<button class="text-white bg-blue-500 p-2 rounded-md" @click="submitForm">Send
Message</button>
</div>
</div>
</template>

```

```
<script>
```

```
import axios from 'axios';
```

```

export default {
  data() {
    return {
      displayName: "",
      email: "",
      phone: "",
      message: "",
      files: null,
      jobPost: {
        title: "Marketing Planner",
        description: {
          txt: "<h5><strong>Job Description :</strong></h5><ul><li>Monitor competition
to identify new industry trends and opportunities for innovation</li><li>Develop marketing
strategies based on customer preferences, business objectives, and available
resources</li><li>Conduct market research to identify potential customers and assess their
needs, interests, and preferences</li><li>Work with design teams to develop marketing
materials such as brochures, ads, and presentations</li><li>Create an advertising budget
that reflects the company's priorities while remaining within the parameters set by upper
management</li><li>Develop an effective communications strategy for each campaign using
several different channels, including social media, email, push notification, print
ads</li><li>Create a detailed timeline of all marketing activities, including production

```

schedules, deadlines for materials delivery, and event dates and locationsProvide feedback to superiors regarding the effectiveness of new campaigns and recommend adjustments",

```
    },
    ptJobSkills: [
      { id: 1101, name: "Minimum bachelor degree in Marketing Management" },
      { id: 1102, name: "Minimum 3-5 years experience as Marketing
Manager/Director in any marketplace industry" },
      { id: 1103, name: "Good time management and project management skill" },
      { id: 1104, name: "Creative communication skill in both verbal and design" },
      { id: 1105, name: "Multi-language abilities in verbal and non-verbal" },
    ],
  },
};
},
methods: {
  async submitForm() {
    try {
      const formData = new FormData();

      formData.append('displayName', this.displayName);
      formData.append('subject', 'Marketing Planner');
      formData.append('email', this.email);
      formData.append('phone', this.phone);
      formData.append('message', this.message);
      formData.append('ptJobApplyType', 'APPLICATION');
      formData.append('isActive', false);

      if (this.file) {
        formData.append('file', this.file);
      }
      const response = await axios.post(
        'https://bti.id/services/btiportalcorems/api/pt-job-posts/no-auth',
        formData,
        {
          headers: {
            'Content-Type': 'multipart/form-data',
          },
        }
      );

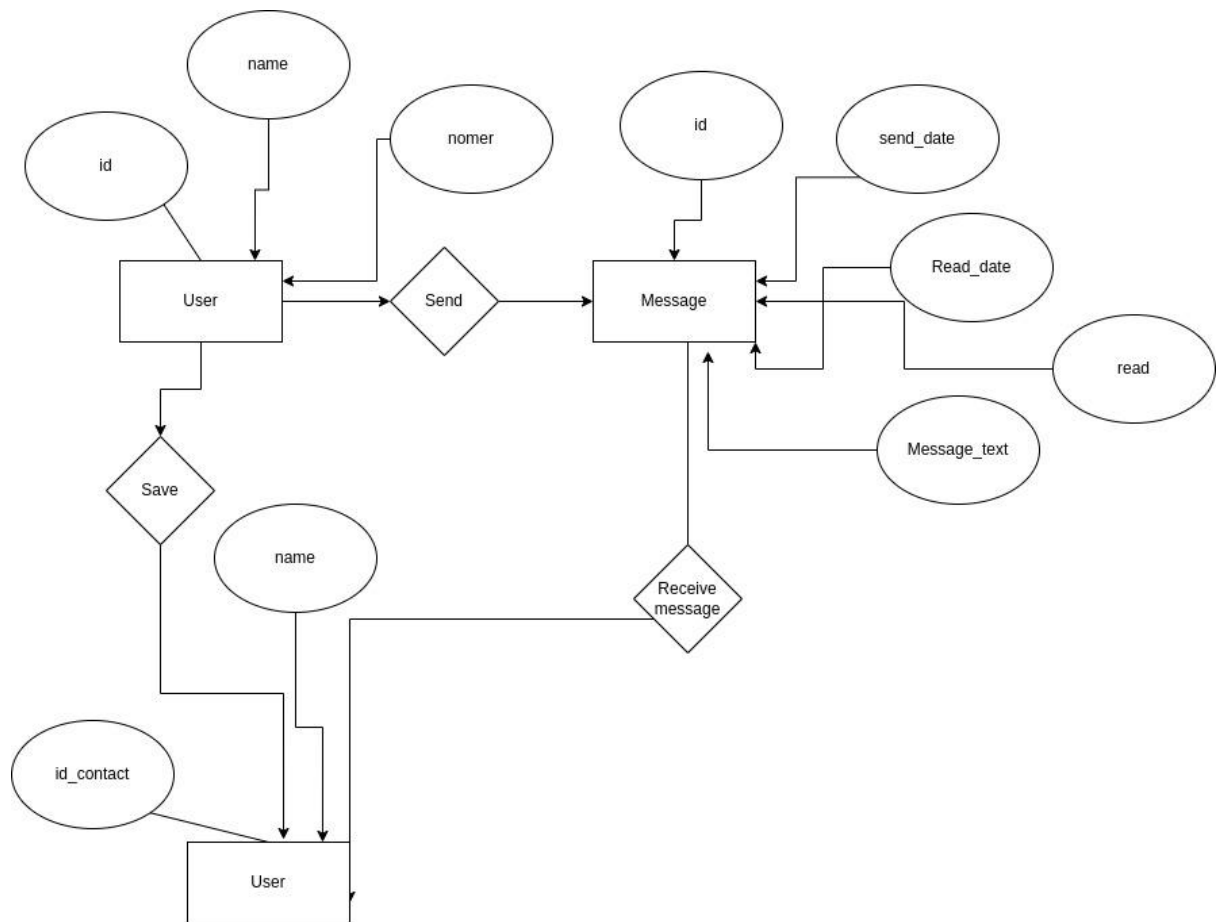
      console.log('Form submitted successfully:', response.data);
    } catch (error) {
      console.error('Error submitting form:', error);
    }
  },
  removeMultiUpload() {
```

```
    },  
    handleFileChange(event) {  
        this.file = event.target.files[0];  
    },  
},  
};  
</script>
```

C.

```
-- Result 1  
SELECT DISTINCT item  
FROM asset;  
  
-- Result 2  
SELECT  
    user_id,  
    GROUP_CONCAT(item ORDER BY item) AS asset  
FROM asset  
GROUP BY user_id;
```

D.



E.

```

1.
function generatePattern(n) {
  let result = "";
  for (let i = 1; i <= n; i++) {
    for (let j = 1; j <= n; j++) {
      if (j <= i) {
        result += j;
      } else {
        result += '*';
      }
    }
    result += '\n';
  }
  return result;
}
  
```

```

console.log(generatePattern(5));
console.log(generatePattern(4));
  
```

2.

```
function findMaxDifference(arr) {  
  let maxDifference = 0;  
  let currentMin = arr[0];  
  
  for (let i = 1; i < arr.length; i++) {  
    const currentNumber = arr[i];  
    const currentDifference = currentNumber - currentMin;  
  
    maxDifference = Math.max(maxDifference, currentDifference);  
    currentMin = Math.min(currentMin, currentNumber);  
  }  
  
  return maxDifference;  
}  
  
// Contoh penggunaan  
const harga = [10, 7, 5, 8, 11, 9, 1];  
console.log(findMaxDifference(harga)); // Output: 6
```

F.

1. Cypress:

Sebuah alat pengujian end-to-end (E2E) yang digunakan untuk menguji aplikasi web secara otomatis.

2. Kubernetes:

Platform open-source untuk otomatisasi deployment, scaling, dan manajemen aplikasi kontainer.

3. ORM (Object-Relational Mapping):

Suatu teknik pemrograman yang mengonversi data antara sistem basis data terstruktur dan format yang dapat digunakan dalam bahasa pemrograman berorientasi objek.

banyak contohnya seperti prisma di node js

4. Class:

Dalam pemrograman berorientasi objek, sebuah blueprint untuk membuat objek, yang mendefinisikan atribut dan perilaku objek.

5. Function:

Sebuah blok kode yang dapat dipanggil dan dijalankan untuk melakukan tugas tertentu.

6. DevOps:

Praktik pengembangan perangkat lunak yang menyatukan pengembangan (Dev) dan operasi (Ops) guna meningkatkan kolaborasi dan pengiriman perangkat lunak secara efisien.

7. Docker:

Platform untuk mengembangkan, mengirim, dan menjalankan aplikasi dalam kontainer.

G.

Microservices Architecture:

Teknologi Terbaik: Kubernetes untuk orkestrasi, Spring Boot untuk backend, React atau Angular untuk frontend.

Alasan:

pengembangan bisa sangat skalabilitas dan fleksibilitas sangat memungkinkan pengembangan di kedepannya

menjadi kan aplikasi sangat failover yang baik jadi jika ada satu layanan gagal yang lain tidak berpengaruh dan tetap bisa berjalan

Serverless Architecture:

contoh: AWS Lambda, Azure Functions, Google Cloud Functions untuk fungsi serverless, dan React atau Angular untuk frontend.

Alasan:

biaya murah bahkan bisa gratis berdasarkan penggunaan jika tidak ada yang digunakan skalabilitas yang otomatis dapat dinaik turun kan sesuai permintaan kapasitasnya

Event-Driven Architecture:

Teknologi Terbaik: Apache Kafka untuk event streaming, Node.js atau Spring Cloud Streams untuk layanan pengolah event.

Alasan:

realtime data processing

Skalabilitas data sangat mudah diatur