

Prepared and Completed by Vaibhav Kansal

Below is the **structured documentation** for your Vue.js component, organized into different parts:

Job Opportunities Component Documentation

1. Introduction

This Vue.js component displays job opportunities in a **carousel-style** scrolling interface. Users can navigate through job postings using **left and right arrow buttons** (on larger screens) or **touch gestures** (on mobile devices). Each job posting displays details like **title, location, date, and salary** with an option to view the **previous year's paper** for the job.

2. Component Structure

The component consists of three main sections:

1. **Header** – Displays the section title.
 2. **Job Cards** – Displays job listings in a horizontally scrollable container.
 3. **Navigation Controls** – Provides left and right arrows to scroll through job listings.
-

3. Template Section (<template>)

3.1. Section Title

```
<h2 class="text-center"><u class="underline">Job Opportunities</u></h2>
```

- Displays the section heading “**Job Opportunities**” in a centered format with an underline.

3.2. Job Cards Wrapper

```
<div class="job-cards-container flex gap-4 overflow-x-auto md:overflow-hidden"
```

```
  ref="jobCardsWrapper"
```

```
  @touchstart="startTouch"
```

```
  @touchmove="moveTouch">
```

- The job cards are wrapped inside a **scrollable container**.
- Supports **touch gestures** (@touchstart and @touchmove) for smooth navigation on mobile devices.

3.3. Job Cards (v-for Loop)

```

<div v-for="(job, index) in visibleJobs" :key="index" class="job-card">

  <h3 class="text-lg font-bold">{{ job.title }}</h3>

  <div class="job-details">

    <p class="text-gray-600 flex items-center"><span>📍</span> {{ job.location }}</p>

    <p class="date-text flex items-center"><span>📅</span> {{ job.date }}</p>

    <p class="salary-text flex items-center"><span>💰</span> {{ job.salary }}</p>

  </div>

  <button @click="viewPreviousYearPaper(job)" class="button-text">

    📄 View Previous Year Paper

  </button>

</div>

```

- **Loops through visibleJobs** to display **job title, location, date, and salary**.
- Includes a button to **view previous year's paper** for each job.

3.4. Navigation Buttons (Arrow Controls)

```

<button @click="scrollLeft" class="arrow-btn left-arrow hidden md:flex">

  <i class="arrow left"></i>

</button>

<button @click="scrollRight" class="arrow-btn right-arrow hidden md:flex">

  <i class="arrow right"></i>

</button>

```

- Provides **left and right navigation buttons** for desktop users.
- **Hidden on mobile (hidden md:flex)** since mobile users can swipe.

4. Script Section (<script>)

4.1. Data Properties (data())

```

data() {
  return {
    jobs: [

```

```

    { title: "Government Clerk SHO", location: "Gujarat Government", date: "03-03-2025",
      salary: "40k to 50k" },

    { title: "BPSC Assistance Vacancy", location: "Patna, Bihar", date: "03-03-2025", salary:
      "50k to 60k" },

    { title: "Bihar Police Services", location: "Patna, Bihar", date: "03-03-2025", salary: "40k
      to 50k" },

    { title: "Railway Engineer", location: "New Delhi", date: "03-03-2025", salary: "60k to
      80k" },

    { title: "UPSC Civil Services", location: "India", date: "03-03-2025", salary: "70k to 90k"
    },

    { title: "Paper 6", location: "India", date: "03-03-2025", salary: "70k to 90k" },

  ],
  currentIndex: 0, // Tracks current position in job carousel
  startX: 0, // Used for touch-based scrolling
};
}

```

- Stores **job listings** in an array.
- `currentIndex` determines which jobs are visible.
- `startX` tracks touch events for mobile scrolling.

4.2. Computed Property (`visibleJobs`)

```

computed: {
  visibleJobs() {
    return this.jobs.slice(this.currentIndex, this.currentIndex + 3);
  }
}

```

- **Dynamically updates the displayed jobs** by slicing the jobs array based on `currentIndex`.
- Ensures only **3 jobs are visible at a time**.

4.3. Methods

4.3.1. Scroll Navigation (scrollLeft and scrollRight)

```
scrollLeft() {  
  if (this.currentIndex > 0) {  
    this.currentIndex--;  
  }  
},  
scrollRight() {  
  if (this.currentIndex < this.jobs.length - 3) {  
    this.currentIndex++;  
  }  
}
```

- Moves the job list **left or right** by adjusting currentIndex.
- Ensures the user **cannot scroll past the first or last job**.

4.3.2. Open Previous Year Paper (viewPreviousYearPaper)

```
viewPreviousYearPaper(job) {  
  const formattedTitle = job.title.replace(/\s+/g, '-').toLowerCase();  
  const url = `https://example.com/previous-year-paper/${formattedTitle}`; // Corrected line  
  window.open(url, '_blank');  
}
```

- Converts job title into a **URL-friendly format**.
- Opens the **previous year paper link in a new tab**.

4.3.3. Touch Events for Mobile Scrolling

```
startTouch(e) {  
  this.startX = e.touches[0].clientX;  
  this.scrollLeft = this.$refs.jobCardsWrapper.scrollLeft;  
},  
moveTouch(e) {  
  const touchMoveX = e.touches[0].clientX;  
  const moveDistance = this.startX - touchMoveX;
```

```
this.$refs.jobCardsWrapper.scrollLeft = this.scrollLeft + moveDistance;
}
```

- Enables **swipe-based scrolling** on mobile devices.
-

5. Styling (<style scoped>)

5.1. Job Card Styling

```
.job-card {
  width: 290px;
  height: 250px;
  background: white;
  padding: 8px;
  box-shadow: 2px 2px 8px rgba(0, 0, 0, 0.1);
  display: flex;
  flex-direction: column;
}
```

- Sets the **size, background, padding, and layout** of job cards.

5.2. Navigation Button Styling

```
.arrow-btn {
  background-color: #052A63;
  padding: 10px;
  border-radius: 50%;
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
}
```

- Styles the **left and right arrow buttons**.

5.3. Responsive Adjustments

```
@media (max-width: 768px) {
  .job-card {
```

```
width: 200px;  
height: 230px;  
}  
}
```

- **Reduces card size** on smaller screens.
-

6. Summary

- **Displays job listings** in a scrollable interface.
- Supports **mouse clicks & touch gestures** for navigation.
- Includes a **dynamic link to previous year's papers**.
- Fully **responsive** and works on **mobile & desktop**.

This documentation provides a clear breakdown of each section in the component.