ScheduleJS Competitive Advantages

- **Scalability:**

Applications' resources and core components always have a limit. In today's data-intensive world, developers can reach this limit even quicker. Out of the box, ScheduleJS empowers its users with cutting-edge technology allowing your application graphics to push this limit further than ever. Graphics built with ScheduleJS benefit from using our cross-platform state-of-the-art HTML Canvas rendering engine, lightweight binary tree data representation, and deeply customizable environment to guarantee your application reaches its goals. Whether you care about user experience, reliability, or aesthetics, ScheduleJS offers a scalable solution for your application graphics.

- **Performance:**

Performance is at the core of ScheduleJS. Historically, our solution has always been used to picture large datasets. Our users need to have their application scale over time and ensure smooth navigation and interactions. Performance becomes a key factor when a large quantity of data is shown on the screen. The main competitive advantage of ScheduleJS resides in its unique drawing engine. Our drawing engine uses a dynamic viewport that will only render what the user currently sees, depending on the timeframe and window size. ScheduleJS will hold the rest of the data in highly optimized structures to ensure a seamless user experience.

- **Renderer Architecture:**

Create specific renderers to picture your data on the screen with infinite possibilities. Renderers can be designed to draw activities, additional layers (on top and behind the graphics and/or rows), and event-based drawings (rich interactions) while offering a well-organized development process. This architecture helps you to optimize your application performance while implementing additional features. Complex application graphics are organized into small development blocks to follow an agile methodology.

- **ScheduleJS API:**

ScheduleJS is the JavaScript version of FlexGanttFX. ScheduleJS is the fourth version of Dirk Lemmermann's DLSC technology. Starting with JavaSwing, the DLSC library evolved to FlexGantt, moved to JavaFX with its FlexGanttFX version, and is now coming to JavaScript with ScheduleJS. The ScheduleJS API makes it possible for your development team to dive deep into our proprietary source code to fine-tune every element's behavior by simply overriding a class or method. After more than 20 years of experience, it is safe to say that we cover almost all the advanced APS/PPS/MES/ERP applications' needs through a well-thought and battle-tested API. We are still working on the library to offer new features and performance improvements. Also, we react quickly to provide developer-friendly solutions to help our users in their journey to implement innovative solutions.

- **Timeline:**

ScheduleJS handles time in a contiguous manner. Whether your users want the bigger picture or details on a succession of activities, the timeline component can communicate with your renderers to adapt the rendering strategy. Zooming operations can be done by default by double-clicking on the timeline, selecting a specific timespan, using a combination of the control key and the mouse wheel, or programmatically.

- **Rich Interactions:**

The combination of the renderer architecture, the ScheduleJS API, and a large set of available events allows the developer to implement rich interaction for any platform. Drag and drop operations are implemented out of the box and offer a way to visually interact with activity data. ScheduleJS can react to any action or information and trigger external API calls to integrate seamlessly into your workflow. Feedback for those operations and their status can be pictured in real-time for all users to promote a single source of truth.