SIH – My contributions

I am very much interested in the frontend and backend and the overall ui.

Also I would like to contribute my part to parameter optimizing and finding the dataset for our ai model

Skillset –

Designnn (por 😉)

Frontend (user interactive)

AI/ml (I know only the ml part not the CNNs and all)

Really looking forward to learn all of the following

React.js + TailwindCSS
These tools are used to build the main user interface for section controllers. React.js allows the system to present real-time data interactively through reusable components such as dashboards, alerts, and train lists. TailwindCSS ensures that the interface is modern, responsive, and consistent without requiring extensive custom styling. Together, they form the primary workspace where controllers view and respond to AI-generated recommendations.

D3.js or Recharts
Data visualization is essential for turning complex scheduling and train traffic data into intuitive graphics. D3.js enables custom visualizations such as train movement timelines, track occupancy charts, and congestion heatmaps. Recharts can be used for simpler metrics like bar or line charts, including throughput and delay summaries. These tools make abstract optimization results easier to interpret and act upon.

Grafana
Grafana provides monitoring dashboards for system performance and health. Instead of visualizing train movements, it displays the performance of the AI engine itself, such as optimization response times, number of conflicts resolved, and overall throughput improvements. Controllers and administrators can use Grafana to ensure the system is functioning reliably and within performance expectations.

Mapbox / Leaflet.js
These libraries are used to create geospatial visualizations of railway networks. Tracks, stations, and trains are displayed on interactive maps in real time. Controllers can see where trains are located, how sections are occupied, and what routing or holding decisions

are being suggested. This helps bridge the gap between optimization models and the physical railway layout.

FastAPI / Flask
FastAPI and Flask act as backend frameworks to expose optimization and prediction models as web services. They allow the frontend to request optimized schedules, fetch train data, and receive conflict resolution suggestions through APIs. These services connect the intelligence layer (optimization, machine learning) with the user interface, acting as translators between the models and the controllers.

Spring Boot
Spring Boot is used where deep integration with legacy railway systems is required. Many existing Train Management Systems or signalling infrastructures run on older technologies. Spring Boot provides robust enterprise-grade tools for connecting to these systems, transforming their data, and exposing it in a form that the new AI system can use. It acts as an adaptor between old infrastructure and new intelligence.

Kong / AWS API Gateway
These tools manage how APIs are exposed to different applications. They enforce authentication, rate limiting, and secure communication. By sitting in front of backend services, they ensure that only authorized systems can access train data and optimization results, and that no single client can overload the system. They act as a traffic controller for data requests, maintaining order and security.

Docker & Kubernetes
Docker packages each component of the system into a portable container, ensuring consistency across development, testing, and deployment. Kubernetes orchestrates these containers in production, managing scaling, load balancing, and fault recovery. Together, they provide the infrastructure backbone that keeps all services running reliably, even under heavy traffic or unexpected disruptions.

Jenkins / GitHub Actions
These tools automate the process of testing, building, and deploying the system. Whenever changes are made to the code, Jenkins or GitHub Actions validate the updates, package the components, and deploy them into the production environment safely. This ensures that the system remains stable and reduces the risk of introducing errors into a live railway environment. They serve as guardians that enforce quality and reliability before any new feature or fix goes live.