A photograph of a railroad crossing at dusk or dawn. In the foreground, a white pole holds two large, circular flashing red lights. The lights are illuminated, and their red glow is visible. In the background, a train with several locomotives and freight cars is approaching the crossing. The train's headlights are on, and its lights are reflecting on the tracks. The sky is dark, and there are some trees and bushes in the background. The overall scene is dimly lit, with the primary light sources being the crossing lights and the train's headlights.

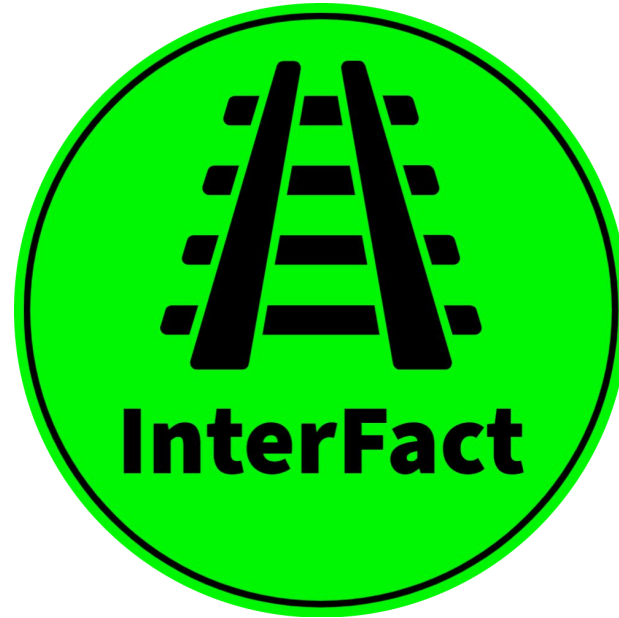
InterFact: Railroad Crossing Information System

InterFact Client & Information:



Huseyin Ergin

You know him



Determines if a train is blocking an intersection in Muncie, using cameras and machine learning



An Admin Dashboard for reviewing data about the InterFact system

4th Iteration Features

- Interfact Admin Dashboard:
 - SQL Database Integration for User Reports
 - Intersection Prediction Data



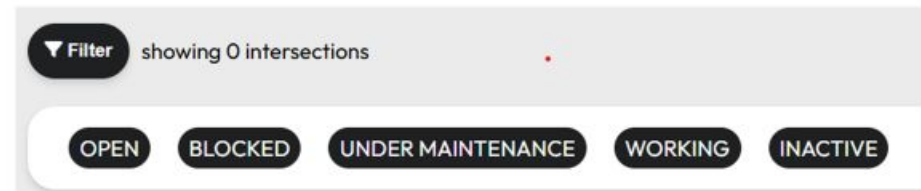
Interfact Iteration 4 Demo!



Camera Maintenance Filter

Intersections can now be sorted
based on which cameras are under
maintenance!

```
case 'Maintenance':  
  if(isFilterOpen !== true || isFilterBlocked !== true){  
    setIsFilterMaintenance(!isFilterMaintenance)  
  }  
  break;
```



Camera Operation Status Filters

- Intersections can now be sorted based on if the cameras are being regularly updated!
 - If data is not received from a specific camera after 10 minutes of inactivity, the camera status is Inactive

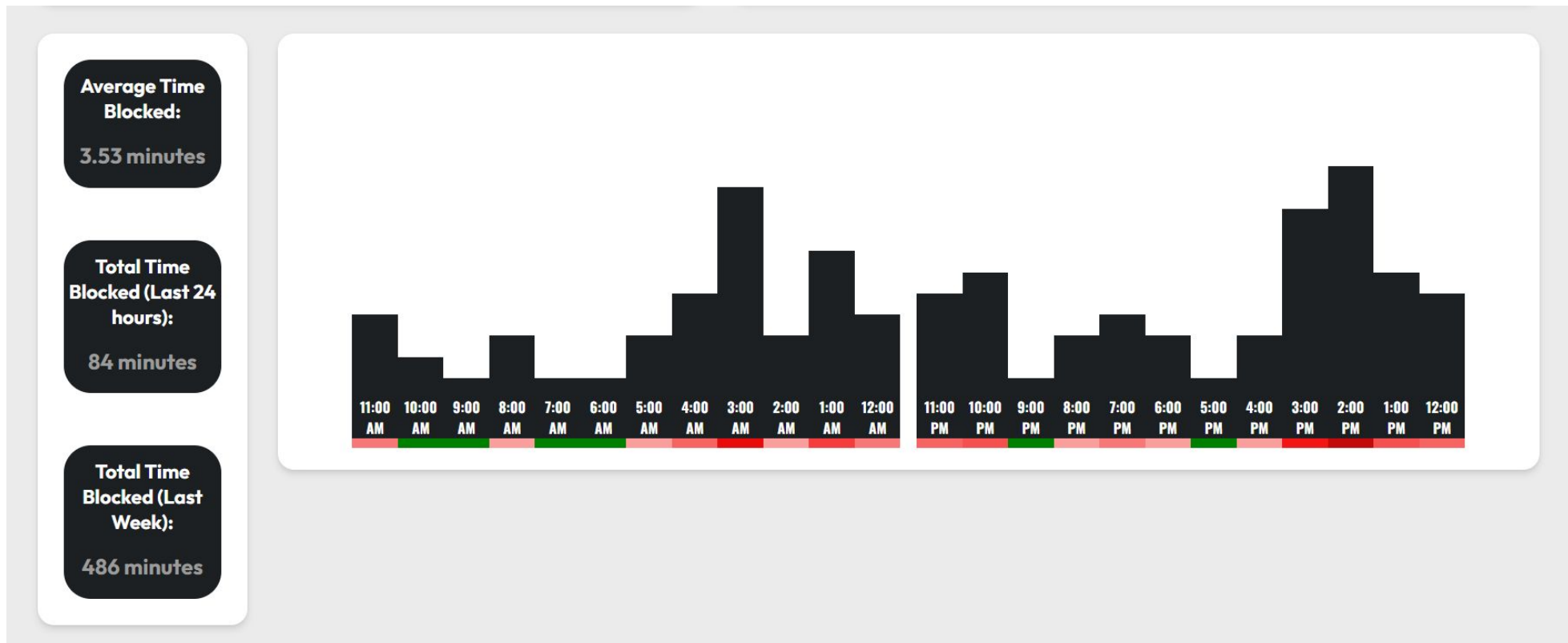


```
// Cameras that have been updated within 10 minutes
if (isFilterWorking){
    if(calculateDifferenceInMinutes(item.timestamp) < 10){
        return item.status === "OPERATIONAL"
    }
}

// Cameras that have not been updated within 10 minutes
if (isFilterNotWorking){
    if(calculateDifferenceInMinutes(item.timestamp) > 10){
        return item.status === "INACTIVE"
    }
}
```

Ethan

I worked on creating several data analysis widgets.




Brooke

File	% Stmts	% Branch	% Funcs	% Lines
-----	-----	-----	-----	-----
All files	73	67.64	75	72.75

code coverage

I worked on testing the new features as well as fixing bugs the new features introduced.



Timestamp | December 2, 2024 at 05:34:00PM UTC-4

Reports Received 1

Log ID:1308

Camera ID:HAC1

Timestamp:3/5/2025, 12:29:22 PM

Filename:ITSNUMBERS.JPG

Status:CLOSED

Path:/path/to/imageFILLER.jpg

Confirm or deny report:

👍

👎

Camera Logs

Log ID:240

Camera ID:ELL1

Timestamp:11/6/2024, 3:55:00 AM

Filename:3343330240_20241106035500_IMAG1240-100-1240.JPG

Status:OPEN

Path:/path/to/image240.jpg

Log ID:239

Camera ID:ELL1

Timestamp:11/6/2024, 3:50:00 AM

Filename:3343330239_20241106035000_IMAG1239-100-1239.JPG

Status:OPEN

Path:/path/to/image239.jpg

Log ID:238

bug showing reports for the wrong intersection

Tanner

I worked on making a transition when clicking on the links on top of the page to add more to the user experience.

```
<body ref={container}>
  <div className='transition'>
    <div className='transition-row row-1'>
      {[...Array(5)].map((_, i) => (
        <div key={i} className='block'></div>
      ))}
    </div>
    <div className='transition-row row-2'>
      {[...Array(5)].map((_, i) => (
        <div key={i} className='block'></div>
      ))}
    </div>
  </div>
</body>
```

```
function animateTransition() {
  return new Promise((resolve) => {
    gsap.set('.block', { visibility: 'visible', scaleY: 0 });
    gsap.to('.block', {
      scaleY: 1,
      duration: 1,
      stagger: {
        each: 0.1,
        from: 'start',
        grid: [2, 5],
        axis: 'x',
      },
    },
    ease: ease,
    onComplete: resolve,
  ));
}

document.querySelectorAll('a').forEach((link) => {
  link.addEventListener('click', (event) => {
    event.preventDefault();
    const href = link.getAttribute('href');

    if (href && !href.startsWith('#') && href !== pathname) {
      animateTransition().then(() => {
        router.push(href);
      });
    }
  });

  revealTransition().then(() => {
    gsap.set('.block', { visibility: 'hidden' });
  });
}, [isMounted, pathname]);

if (isExcluded) {
  return (
    <html lang='en'>
      <body>{children}</body>
    </html>
  );
}
```

Mason

This iteration, I worked on a few small usability features such as researching / using localStorage in Java to save preferences like filter presets. I also started working on implementing a dark theme and dark theme toggle using the same browser storage.

Additionally, I worked on creating the presentation slides and updating the documentation.

```
//----- Save Filter States to localStorage -----  
// Will run whenever a filter is changed  
useEffect(() => {  
  // Store current filter states in localStorage whenever any of the filter state changes  
  const filterStates = {  
    isFiltering,  
    isFilterOpen,  
    isFilterBlocked,  
    isFilterMaintenance,  
    isFilterWorking,  
    isFilterNotWorking  
  };  
  
  // Save the filter states in JSON string  
  localStorage.setItem('filters', JSON.stringify(filterStates));  
}, [isFiltering, isFilterOpen, isFilterBlocked, isFilterMaintenance, isFilterWorking, isFilterNotWorking]);
```

```
// ----- Dark Theme -----  
  
// ----- Set Theme -----  
useEffect(() => {  
  // Check localStorage if the theme was set previously  
  const savedTheme = localStorage.getItem('theme');  
  if (savedTheme === 'dark') {  
    setIsDarkTheme(true);  
    // Apply dark theme to body element  
    // Might need to further specify additions if colors beyond background body are to be changed  
    document.body.classList.add('dark-theme'); // Apply dark theme to body  
  } else {  
    setIsDarkTheme(false); // aka regular theme  
    document.body.classList.remove('dark-theme'); // Remove dark theme from body element  
  }  
  // Empty array below makes sure this runs on load  
}, []);
```

Bella

I worked on the requests page. Visually you see how a request is sorted by street id and how many user have requested that intersection.

Camera Requests

JEF1 4

MON1 3

ELL1 1

```
6
7 export default function requests() {
8
9   const userFeedback = useUserFeedback();
10  const params = useParams();
11  const [requests, setRequests] = useState<string[] | null>([]);
12  const id = Array.isArray(params.id) ? params.id[0] : params.id;
13
14  const getRequests = (id : string): string[] => {
15    return userFeedback.flatMap(user => {
16      if (Array.isArray(user.requests)) {
17        return user.requests.filter((request : string) => {
18          return request === id
19        });
20      }
21      return [];
22    });
23  };
24
25  useEffect(() => {
26    if (userFeedback.length > 0 && id) {
27      const fetchedRequests = getRequests(id);
28      setRequests(fetchedRequests);
29    }
30  }, [userFeedback, id]);
31
```

Mentor Feedback

- Feedback :
Bradley told us the new features look good functionally but the ui design could use a little updating to something more like a bar chart.

He also told us it would be ideal to only render a few logs at a time and have a page system to avoid lagging due to the amount of logs the program was trying to load at one time.



Client Feedback

- Feedback:
 - Would like to transition from normalized score to percentage. i.e. How likely is the intersection going to be blocked at a given time of day?
 - New features are usable and show potential.
- Changes Made:
 - Added notes & changes to Iteration 5 goals.



Planned Iteration 5 Features:

- More detailed prediction display data
 - Transition normalized score to blocked percentage
 - Break up single 24 hour visual into visuals for each day of the week
- Implement catch for average calculation for missing log reports.

