Fuel Log

MERN Stack Application

By: Joshua Phillips, Jordan Saker

About the application

The fuel usage data from vehicle assets can provide useful analytical trends for a company so that it can better manage fuel costs over the running service periods of company assets.

Features

- Graphing and tabular reports
- Multi-vehicle tracking
- Trip and fuel usage monitoring
- Individual employee monthly usage report
- Simple and intuitive UI

Purpose

To help keep track of the fuel usages for a company, a fast and easy to use application that is centered around logging instances of fuel refilling of a vehicle can give assistance to determining how much fuel different company assets require.

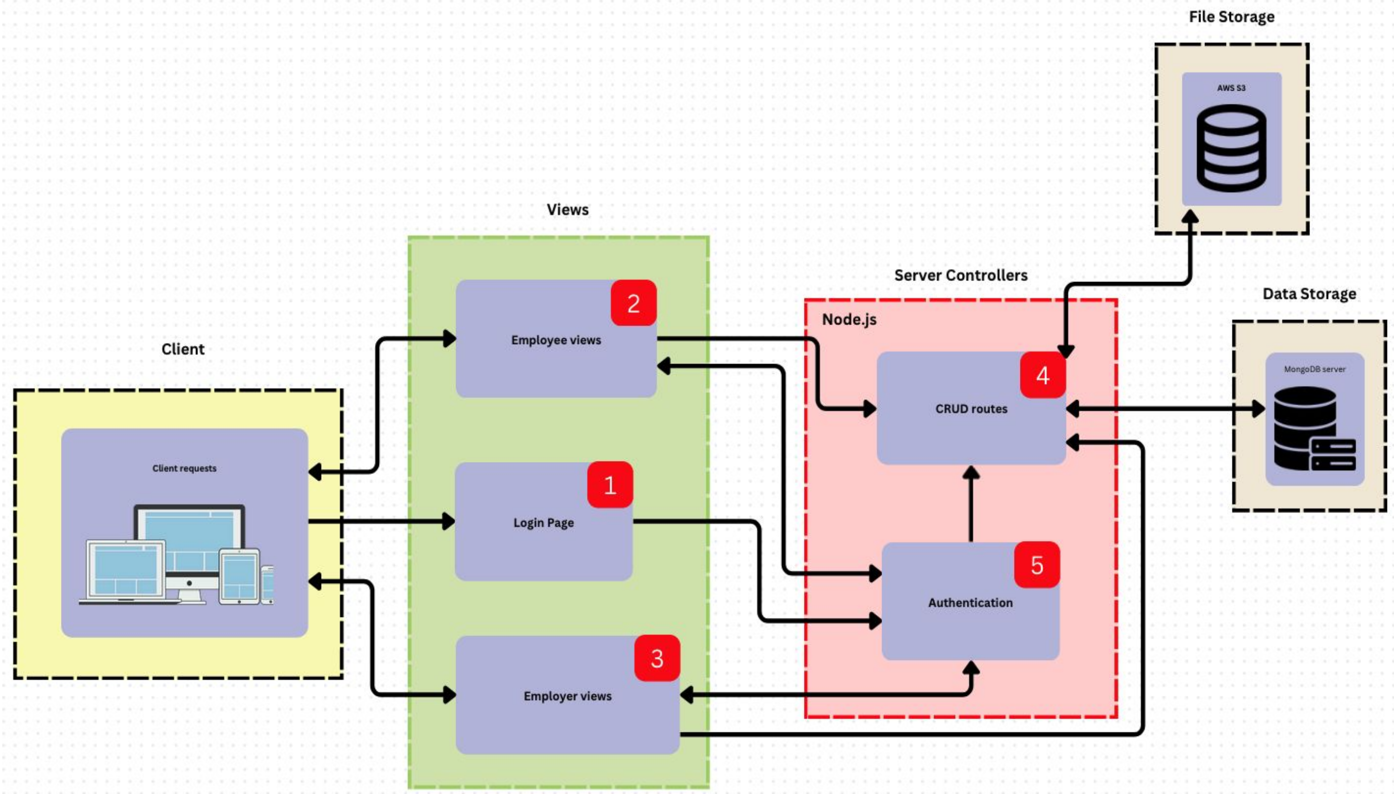
The application uses the recorded data to generate useful reports for the company in an easy-to-understand way, allowing for the company to potentially have the ability to budget and plan for fuel expenses.

Project timeline

- Back-end:
  - Collection modelling. **Duration: 1 day**
  - Basic end-point routes for CRUD operations. **Duration: 1 day**
  - Analytical routes for Logs and Vehicles. **Duration: 1-2 days**
  - Authentication of routes. **Duration: 1 day**
  - Testing and Deployment. **Duration: 1 day**
- Database and storage setup. **Duration: 0.5 day**
- Front-end:
  - Component building. **Duration: 2-3 days**
  - Graphing and Tabular report Components. **Duration: 1 day**
  - Styling. **Duration: 1-2 days**
  - Testing and Deployment. **Duration: 1 day**

Application Architecture

There are two different storage systems that were chosen for the application. A database for storing all the recorded data and a file storage system for storing the vehicle images that will be uploaded for each vehicle.

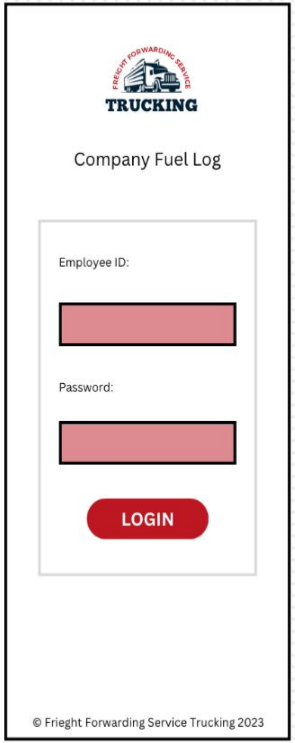


- Access login page, enters details
- Authentication routes query database for password match
- User can access different dashboard depending on account

Mobile Wireframes

The employee pages don’t have a Nav bar. The Nav bar is replaced with the square buttons at the bottom of the page. This is because the employee only uses the app to record a fuel log.

Login Page



Company Fuel Log

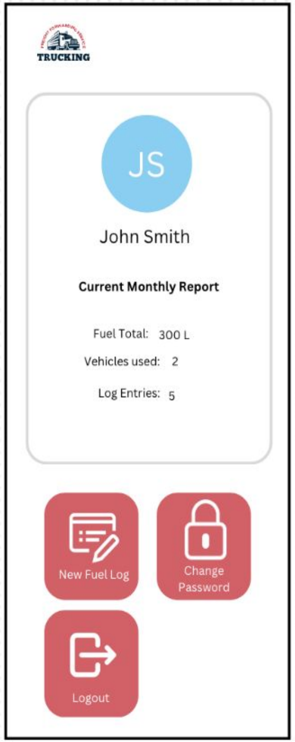
Employee ID:

Password:

LOGIN

© Frieght Forwarding Service Trucking 2023

Employee Home Page



John Smith

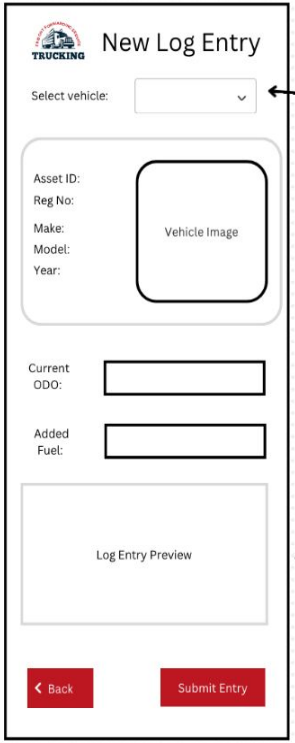
Current Monthly Report

Fuel Total: 300 L

Vehicles used: 2

Log Entries: 5

Log record form



New Log Entry

Select vehicle:

Asset ID:
Reg No:
Make:
Model:
Year:

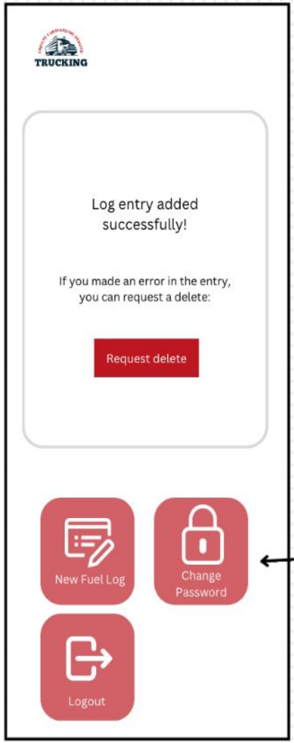
Current ODO:

Added Fuel:

Will trigger a fetch which will fetch the selected vehicle

Component renders after fetch with vehicle details

As the user inputs the log details, details will be rendered here



Log entry added successfully!

If you made an error in the entry, you can request a delete:

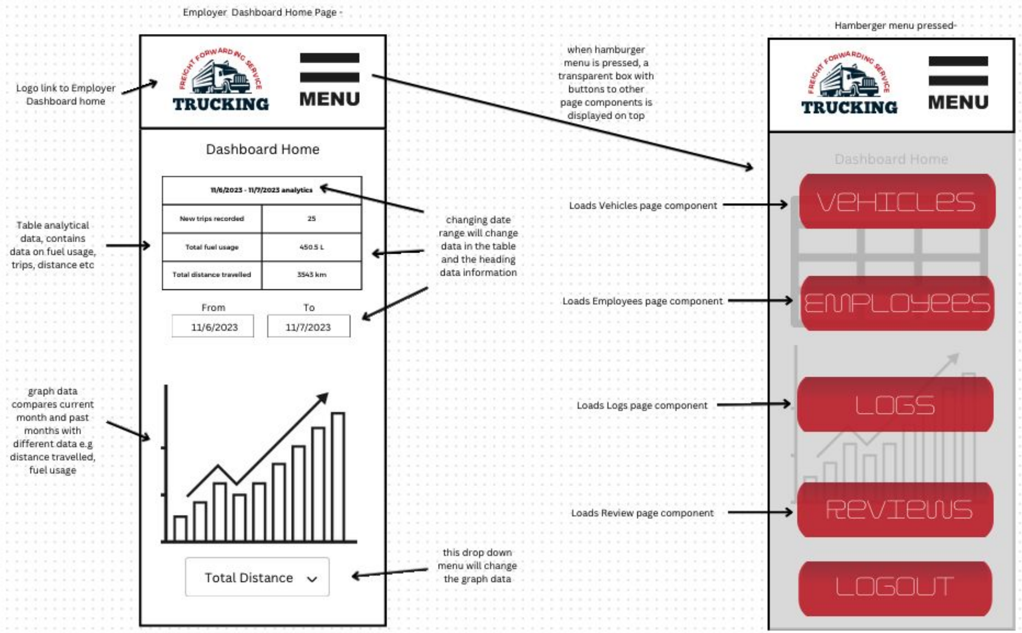
Employ cannot delete a log entry, however they can request a delete on the latest log entry

A form will pop up with "current password" and "new password" fields and a Reset button

Mobile Wireframes cont.

The employer home page was chosen as the page which displayed the analytical graphs and data. Since one of the main features of this application is to do so, it was decided that the reports should be the first thing that's seen and not displayed on a separate page. This could also expanded into its own page once the data grows and allows for more detailed reports, while leave a summarised report on the home page.

Employer dashboard page components



Employer Dashboard Home Page -

Hamberger menu pressed-

Logo link to Employer Dashboard home

Table analytical data, contains data on fuel usage, trips, distance etc

graph data compares current month and past months with different data e.g distance travelled, fuel usage

changing date range will change data in the table and the heading data information

this drop down menu will change the graph data

when hamburger menu is pressed, a transparent box with buttons to other page components is displayed on top

Loads Vehicles page component

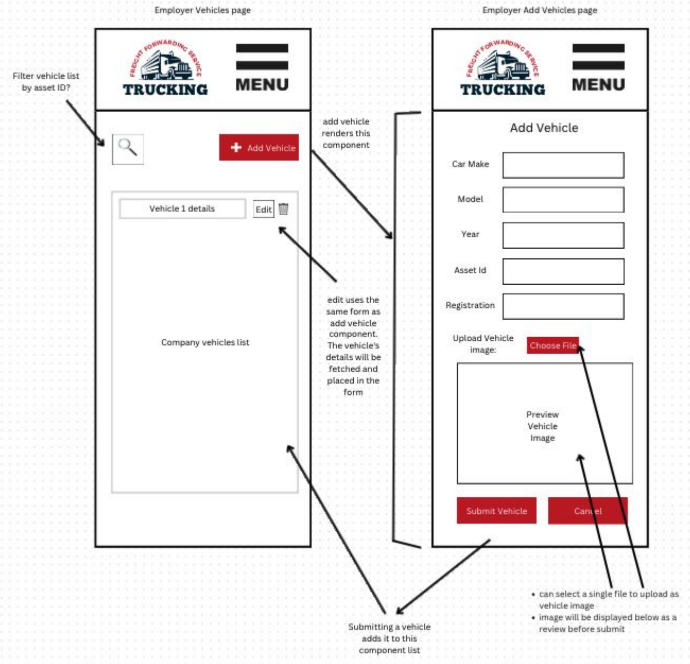
Loads Employees page component

Loads Logs page component

Loads Review page component

11/6/2023 - 11/7/2023 analytics	
New trips recorded	25
Total fuel usage	450.5 L
Total distance travelled	3543 km

Employer vehicle page components



Employer Vehicles page

Employer Add Vehicles page

Filter vehicle list by asset ID?

add vehicle renders this component

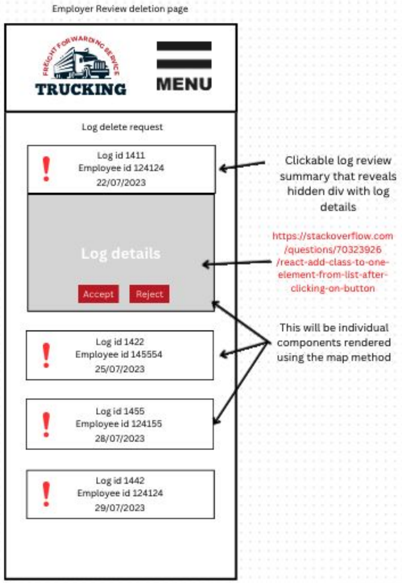
edit uses the same form as add vehicle component. The vehicle's details will be fetched and placed in the form

Submitting a vehicle adds it to this component list

- can select a single file to upload as vehicle image
- image will be displayed below as a review before submit

Mobile Wireframes cont.

Employer review page components

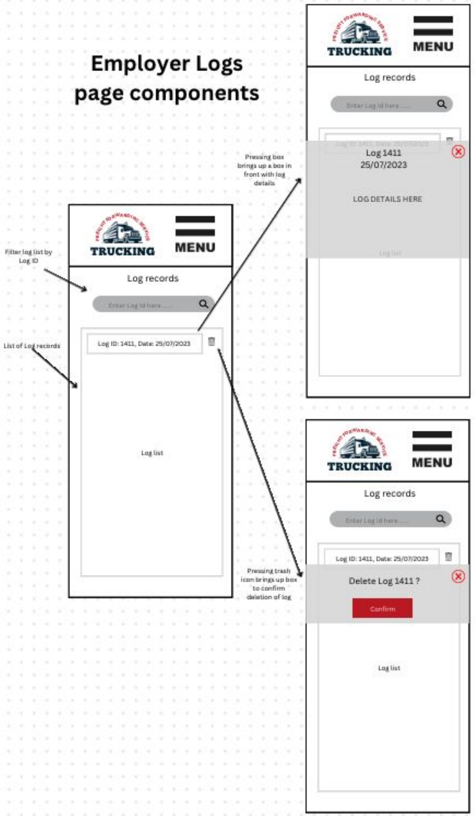


Clickable log review summary that reveals hidden div with log details

https://stackoverflow.com/questions/70323926/react-add-class-to-one-element-from-list-after-clicking-on-button

This will be individual components rendered using the map method

Employer Logs page components



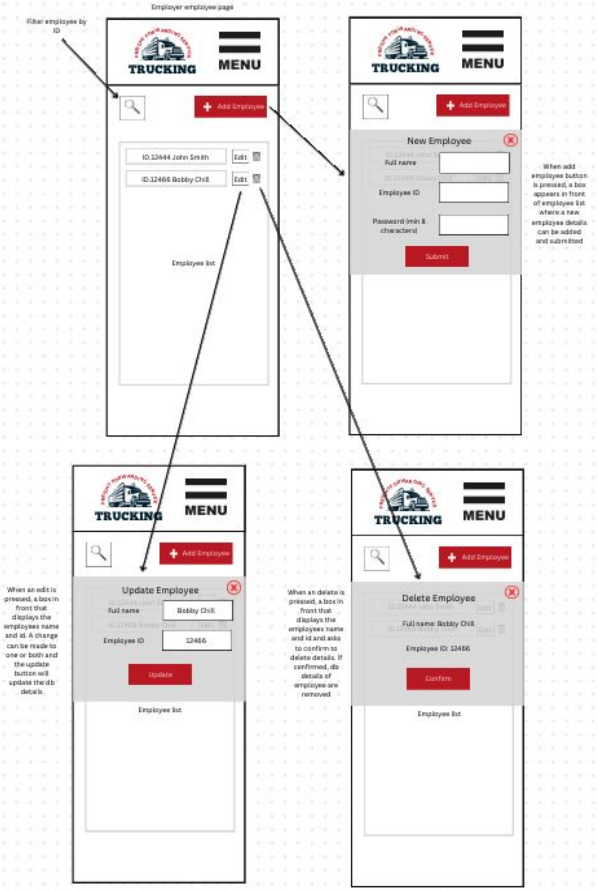
Filter log list by Log ID

List of Log records

Pressing box brings up a box in front with log details

Pressing trash icon brings up box to confirm deletion of log

Employer employee page components



Filter employee by ID

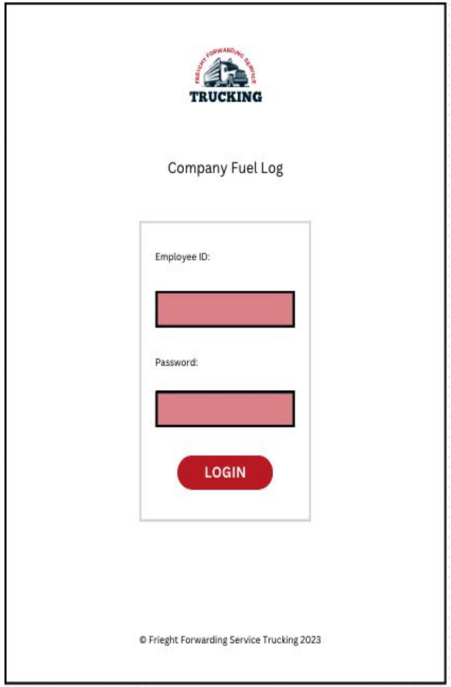
When add employee button is pressed, a box appears in front of employee list where a new employee details can be added and submitted

When an edit is pressed, a box in front that displays the employees name and id. A change can be made to one or both and the update button will update the db details.

When an delete is pressed, a box in front that displays the employees name and id and asks to confirm to delete details. If confirmed, db details of employee are removed

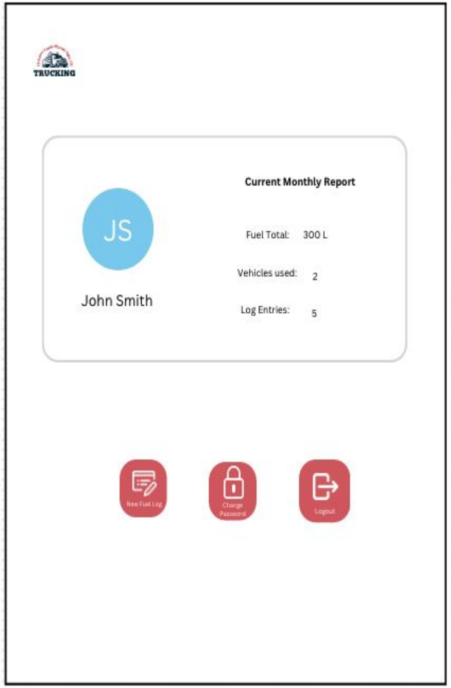
Tablet Wireframes

Login Page



© Frieght Forwarding Service Trucking 2023

Employee Home Page

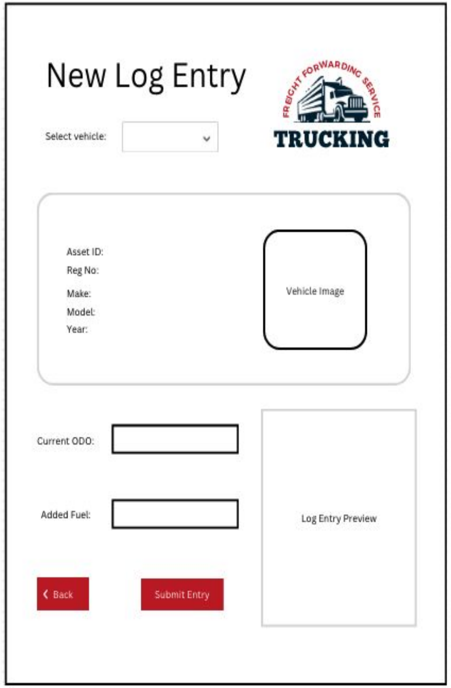


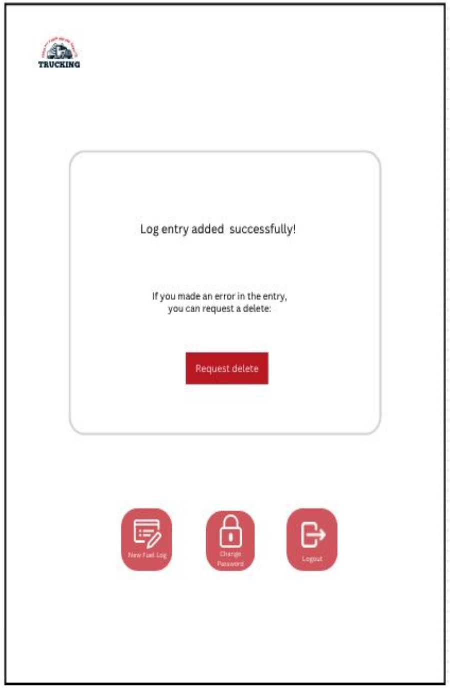
Fuel Total: 300 L

Vehicles used: 2

Log Entries: 5

Log record form





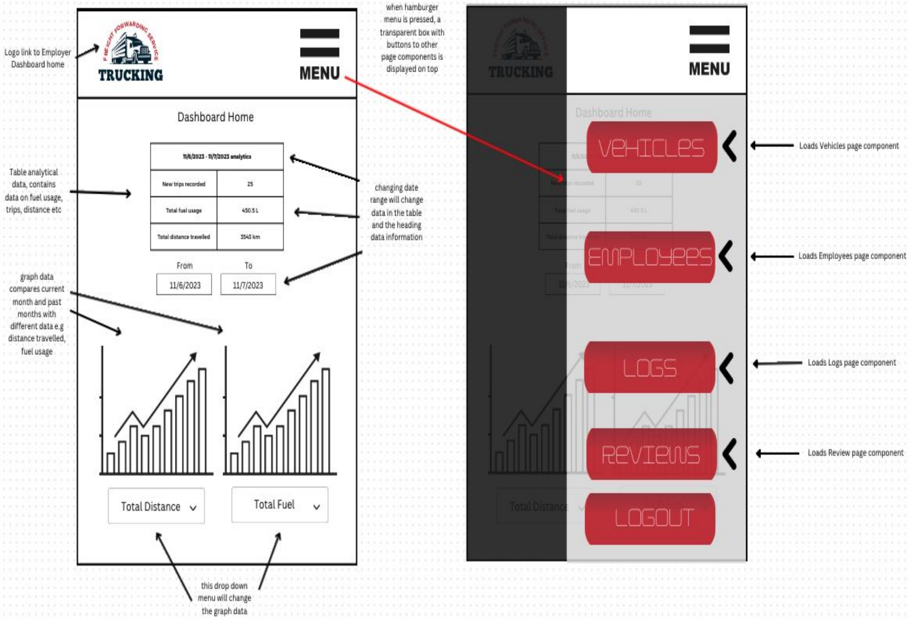
Log entry added successfully!

If you made an error in the entry, you can request a delete:

In tablet view, layout mostly the same except more information displays on dashboard and vehicles displayed across two columns

Tablet Wireframes cont.

Employer dashboard page components



11/6/2023 - 11/7/2023 analytics	
New trips recorded	25
Total fuel usage	450.5 L
Total distance travelled	3543 km

Logo link to Employer Dashboard home

Table analytical data, contains data on fuel usage, trips, distance etc

graph data compares current month and past months with different data e.g distance travelled, fuel usage

when hamburger menu is pressed, a transparent box with buttons to other page components is displayed on top

changing date range will change data in the table and the heading data information

this drop down menu will change the graph data

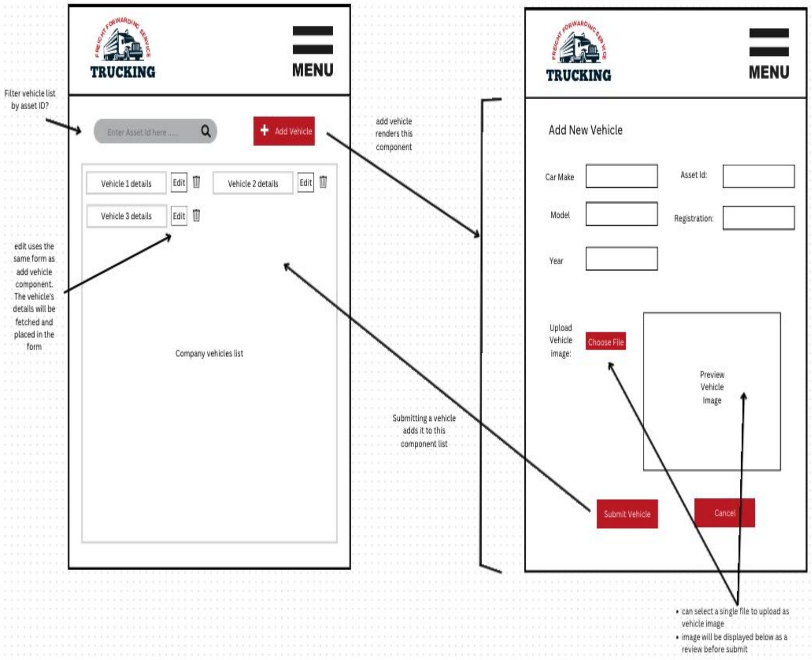
Loads Vehicles page component

Loads Employees page component

Loads Logs page component

Loads Review page component

Employer vehicle page components



Filter vehicle list by asset ID?

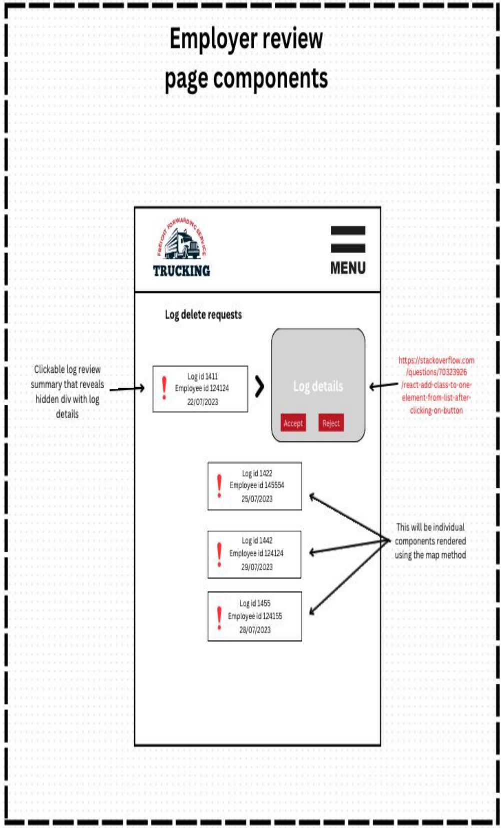
edit uses the same form as add vehicle component. The vehicle's details will be fetched and placed in the form

add vehicle renders this component

Submitting a vehicle adds it to this component list

- can select a single file to upload as vehicle image
- image will be displayed below as a review before submit

Tablet Wireframes cont.

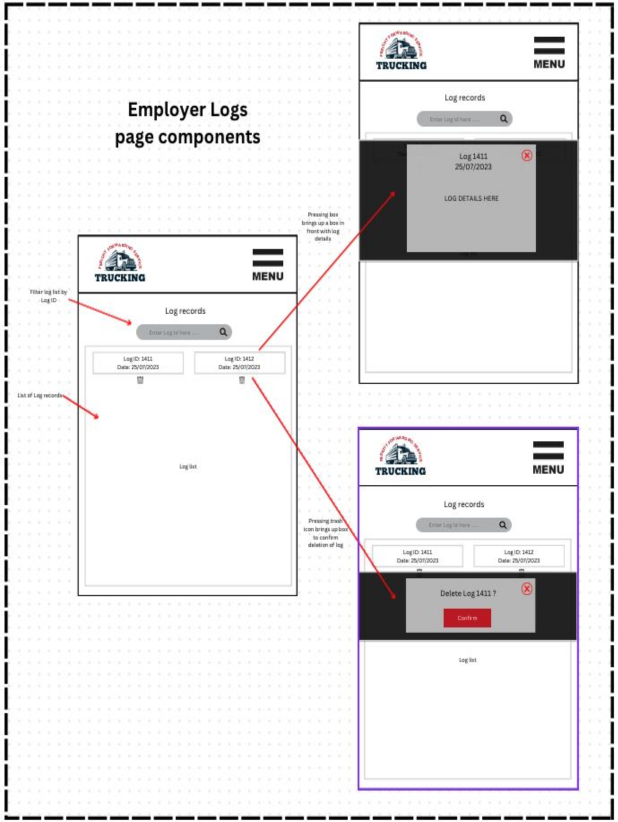


Employer review page components

Clickable log review summary that reveals hidden div with log details

https://stackoverflow.com/questions/70323926/react-add-class-to-one-element-from-list-after-clicking-on-button

This will be individual components rendered using the map method



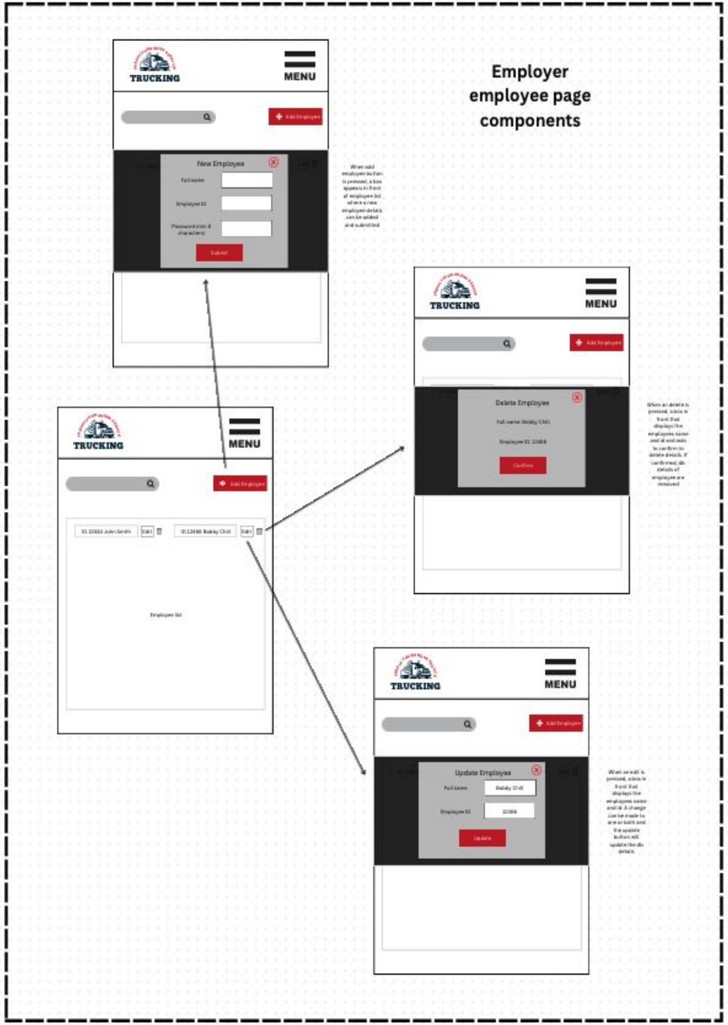
Employer Logs page components

Filter log list by Log ID

List of Log records

Pressing box brings up a box in front with log details

Pressing trash icon brings up box to confirm deletion of log



Employer employee page components

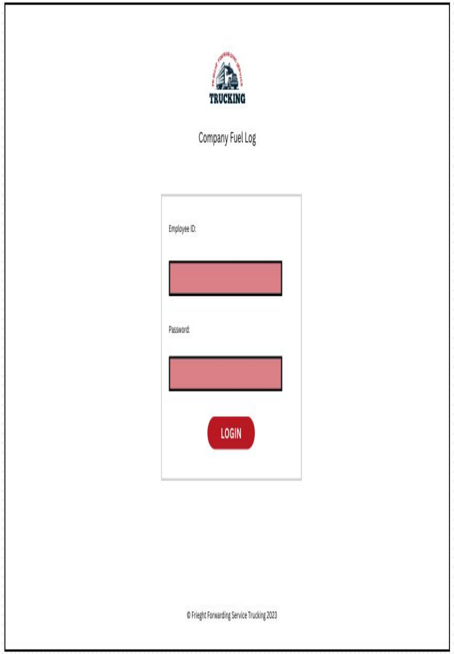
When add employee button is pressed, a box appears in front of employee list where a new employee details can be added and submitted

When an delete is pressed, a box in front that displays the employees name and id and asks to confirm to delete details. If confirmed, db details of employee are removed

When an edit is pressed, a box in front that displays the employees name and id. A change can be made to one or both and the update button will update the db details.

Desktop Wireframes

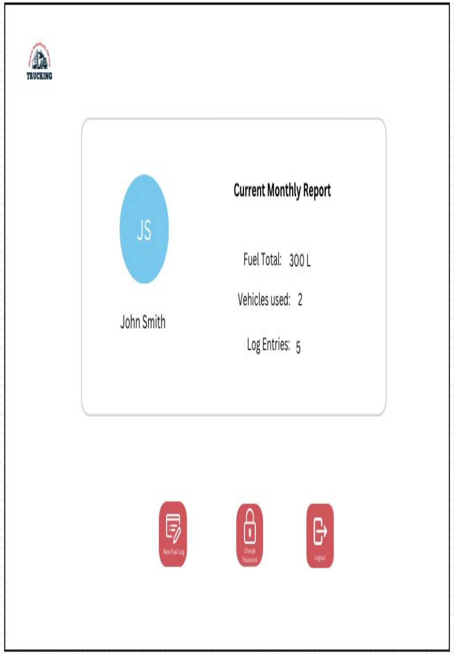
Login Page



Company Fuel Log

© Frieght Forwarding Service Trucking 2023

Employee Home Page



Current Monthly Report

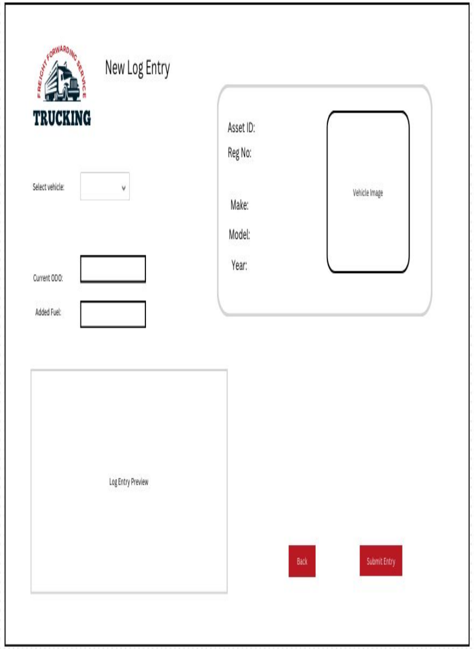
John Smith

Fuel Total: 300 L

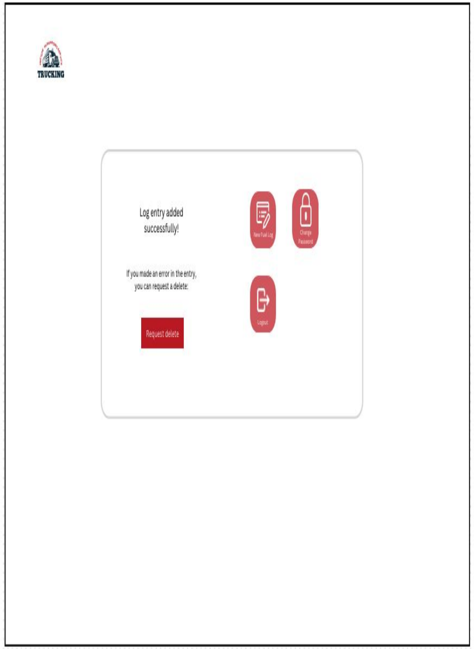
Vehicles used: 2

Log Entries: 5

Log record form



New Log Entry

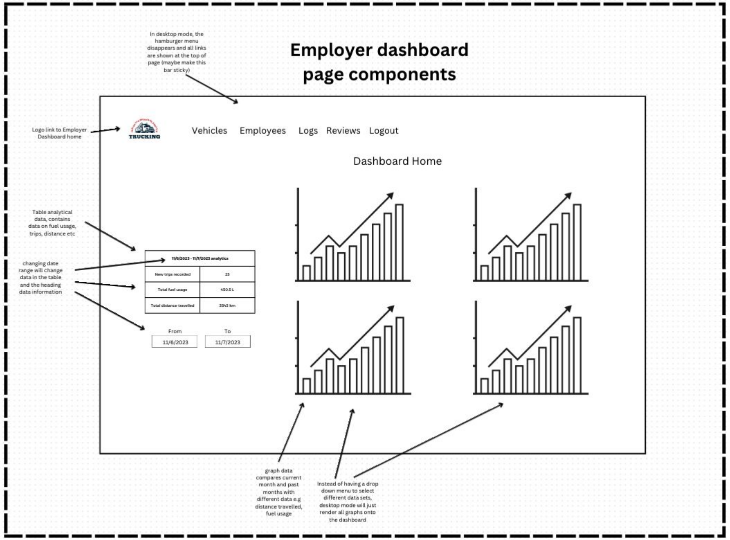


Log entry added successfully!

If you made an error in the entry, you can request a delete:

Desktop Wireframes cont.

In desktop mode, hamburger menu disappears and instead displays all link at top side by side. Dashboard can also display all graphs in desktop mode



Employer dashboard page components

In desktop mode, the hamburger menu disappears and all links are shown at the top of page (maybe make this bar sticky)

Logo link to Employer Dashboard home

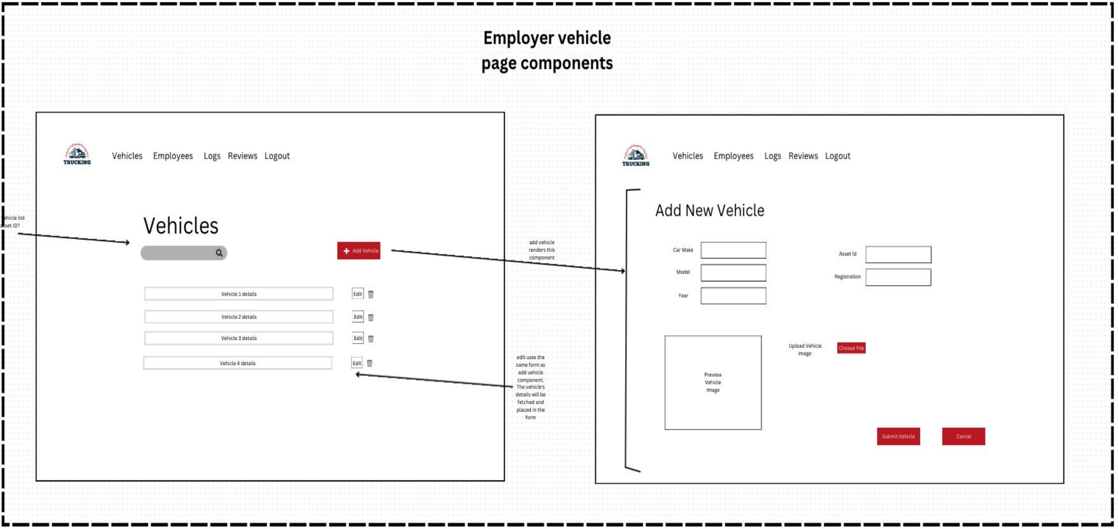
Table analytical data, contains data on fuel usage, trips, distance etc

changing date range will change data in the table and the heading data information

11/6/2023 - 11/7/2023 analytics	
New trips recorded	25
Total fuel usage	450.5 L
Total distance travelled	3543 km

graph data compares current month and past months with different data e.g distance travelled, fuel usage

Instead of having a drop down menu to select different data sets, desktop mode will just render all graphs onto the dashboard



Employer vehicle page components

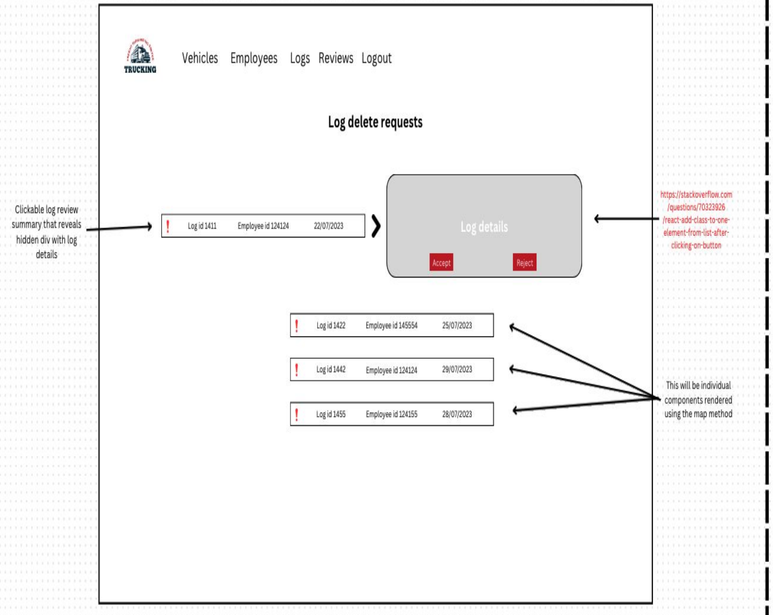
Filter vehicle list by asset ID?

add vehicle renders this component

edit uses the same form as add vehicle component. The vehicle's details will be fetched and placed in the form

Desktop Wireframes cont.

Employer review page components

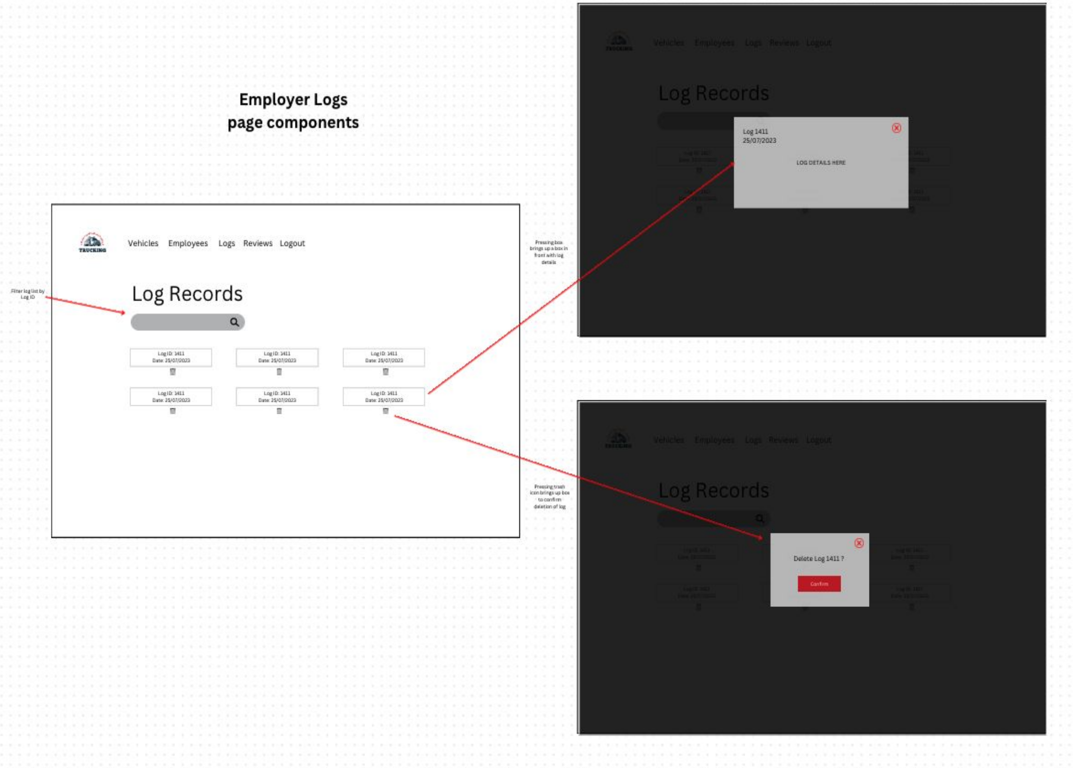


Clickable log review summary that reveals hidden div with log details

https://stackoverflow.com/questions/70323926/react-add-class-to-one-element-from-list-after-clicking-on-button

This will be individual components rendered using the map method

Employer Logs page components

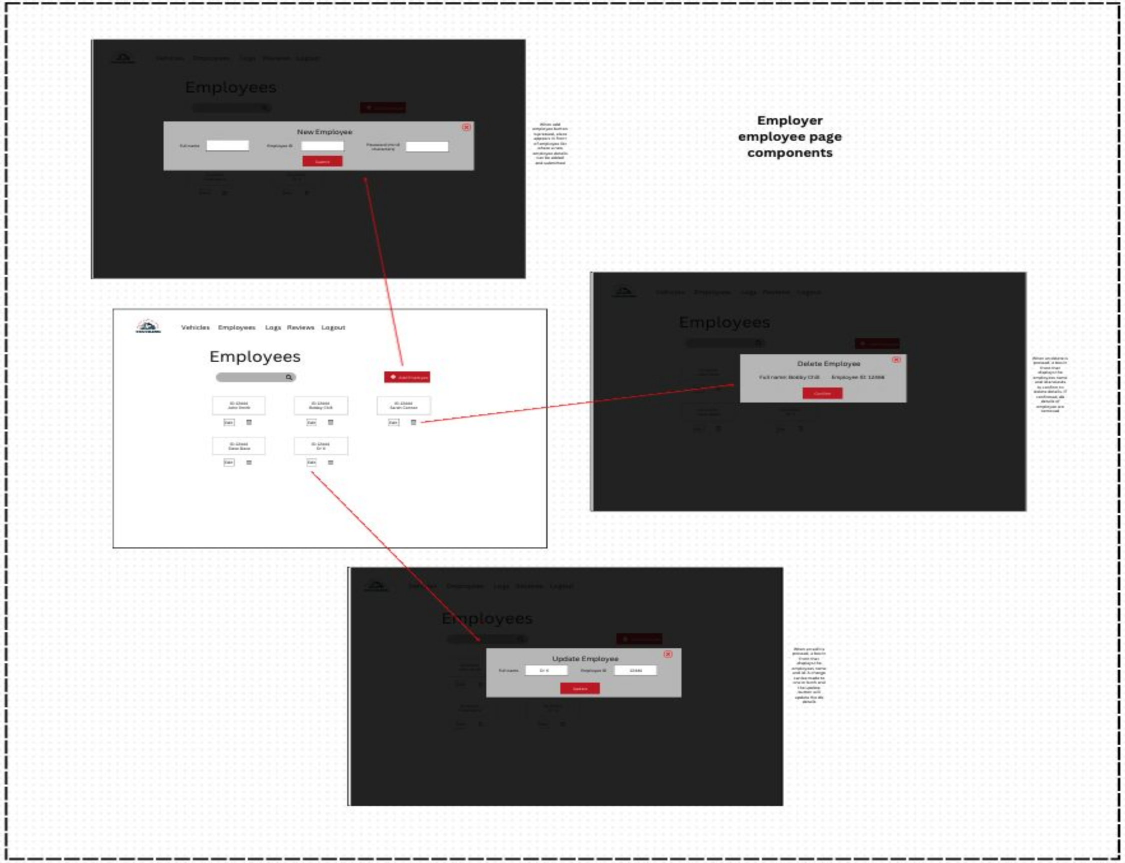


Filter log list by Log ID

Pressing box brings up a box in front with log details

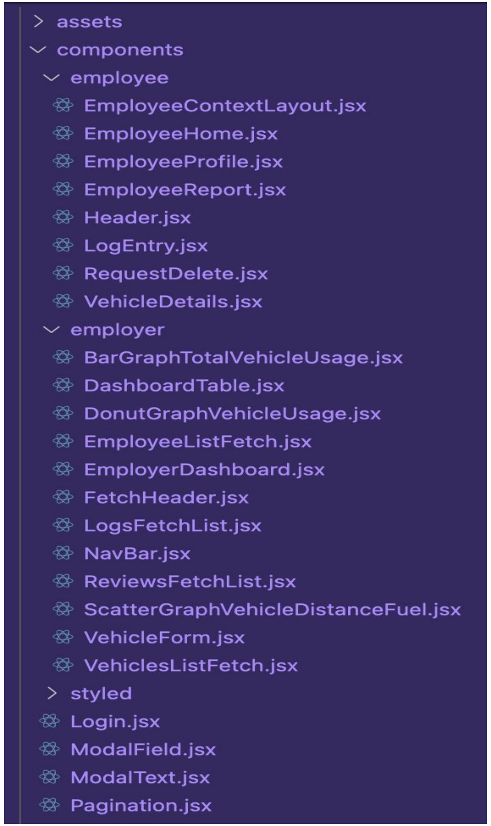
Pressing trash icon brings up box to confirm deletion of log

Desktop Wireframes cont.



Employer employee page components

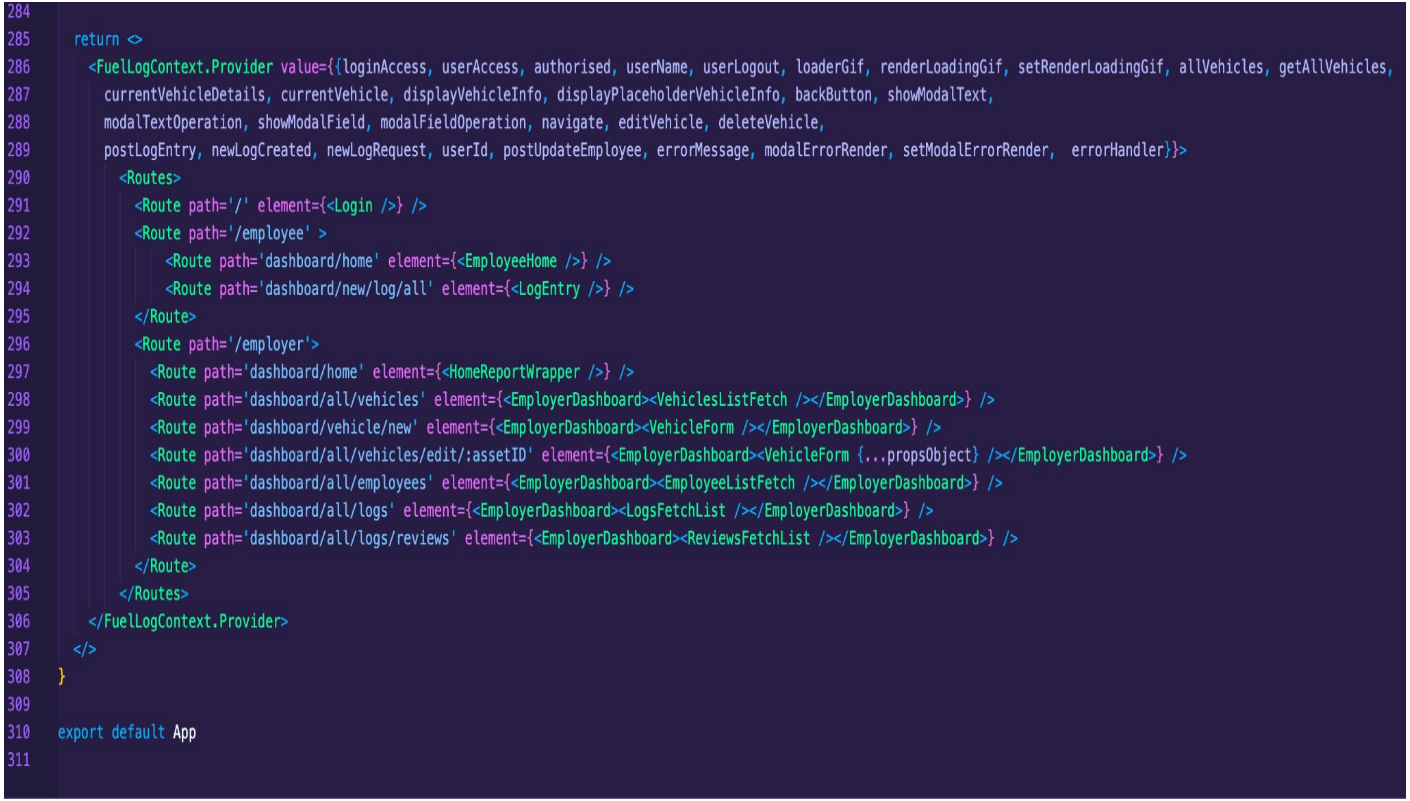
Front end component lists



```
> assets
v components
  v employee
      EmployeeContextLayout.jsx
      EmployeeHome.jsx
      EmployeeProfile.jsx
      EmployeeReport.jsx
      Header.jsx
      LogEntry.jsx
      RequestDelete.jsx
      VehicleDetails.jsx
  v employer
      BarGraphTotalVehicleUsage.jsx
      DashboardTable.jsx
      DonutGraphVehicleUsage.jsx
      EmployeeListFetch.jsx
      EmployerDashboard.jsx
      FetchHeader.jsx
      LogsFetchList.jsx
      NavBar.jsx
      ReviewsFetchList.jsx
      ScatterGraphVehicleDistanceFuel.jsx
      VehicleForm.jsx
      VehiclesListFetch.jsx
  > styled
    Login.jsx
    ModalField.jsx
    ModalText.jsx
    Pagination.jsx
```

- Front end interactiveness of the web application was mainly created by REACT which uses components. These components make it so html can be reused in multiple parts of the application
- Divided into two sections, Employee and Employer components. Other components were also made for styling purposes

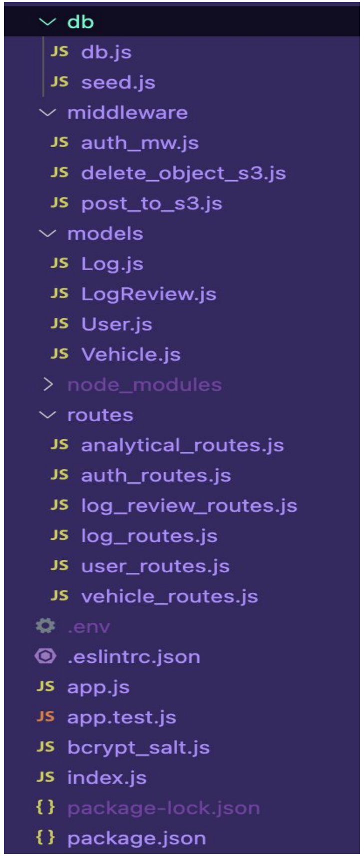
Front end routing



```
  return <>
    <FuelLogContext.Provider value={{loginAccess, userAccess, authorised, userName, userLogout, loaderGif, renderLoadingGif, setRenderLoadingGif, allVehicles, getAllVehicles,
      currentVehicleDetails, currentVehicle, displayVehicleInfo, displayPlaceholderVehicleInfo, backButton, showModalText,
      modalTextOperation, showModalField, modalFieldOperation, navigate, editVehicle, deleteVehicle,
      postLogEntry, newLogCreated, newLogRequest, userId, postUpdateEmployee, errorMessage, modalErrorRender, setModalErrorRender,  errorHandler}}>
        <Routes>
          <Route path='/' element={<Login />} />
          <Route path='/employee' >
              <Route path='dashboard/home' element={<EmployeeHome />} />
              <Route path='dashboard/new/log/all' element={<LogEntry />} />
          </Route>
          <Route path='/employer'>
            <Route path='dashboard/home' element={<HomeReportWrapper />} />
            <Route path='dashboard/all/vehicles' element={<EmployerDashboard><VehiclesListFetch /></EmployerDashboard>} />
            <Route path='dashboard/vehicle/new' element={<EmployerDashboard><VehicleForm /></EmployerDashboard>} />
            <Route path='dashboard/all/vehicles/edit/:assetID' element={<EmployerDashboard><VehicleForm {...propsObject} /></EmployerDashboard>} />
            <Route path='dashboard/all/employees' element={<EmployerDashboard><EmployeeListFetch /></EmployerDashboard>} />
            <Route path='dashboard/all/logs' element={<EmployerDashboard><LogsFetchList /></EmployerDashboard>} />
            <Route path='dashboard/all/logs/reviews' element={<EmployerDashboard><ReviewsFetchList /></EmployerDashboard>} />
          </Route>
        </Routes>
    </FuelLogContext.Provider>
  </>
}

export default App
```

- Majority of routing is performed in App.jsx in src folder
- All components in these routes rely heavily on reducers and contexts to provide global variables for different components. This was chosen to reduce prop usage on components themselves to keep code more readable

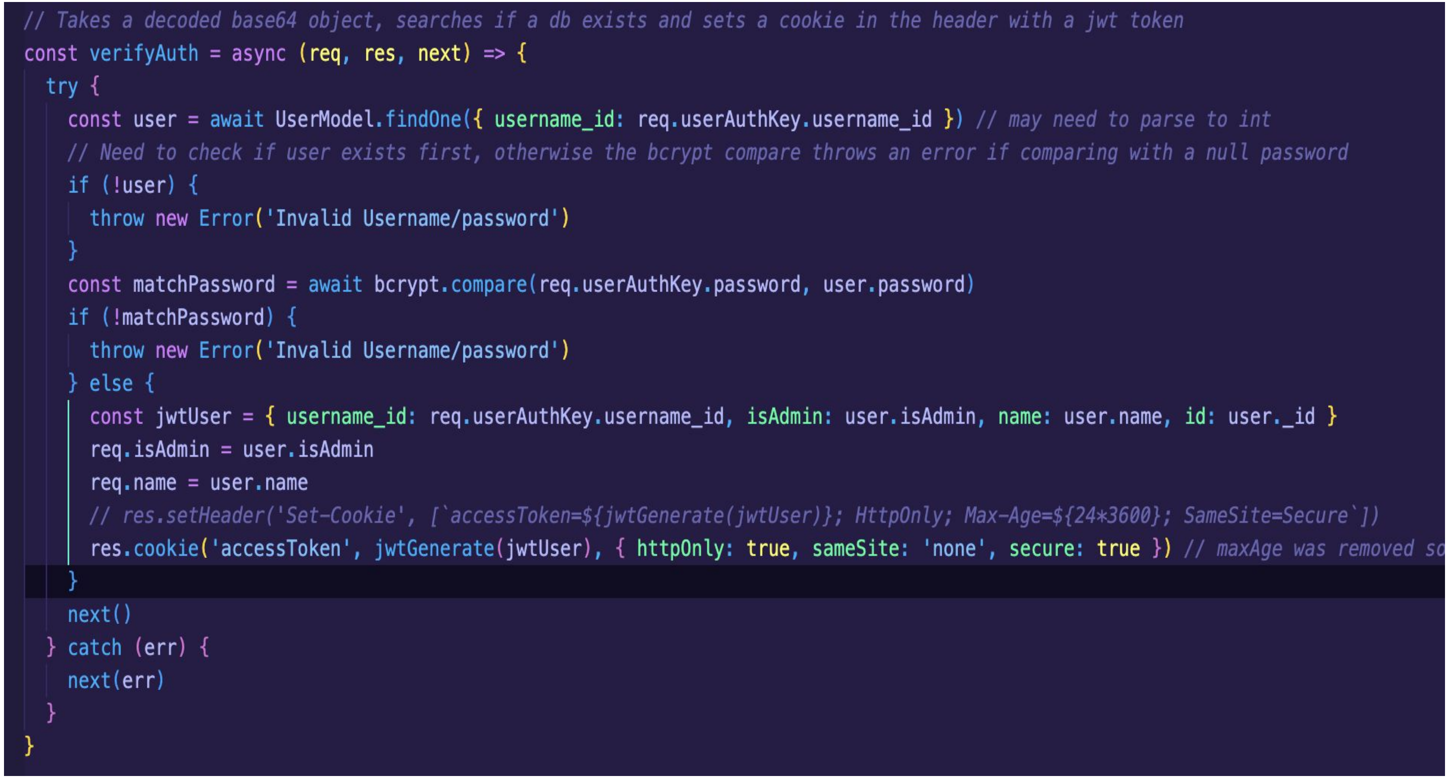
Back end - Express file structure



```
db
  db.js
  seed.js
middleware
  auth_mw.js
  delete_object_s3.js
  post_to_s3.js
models
  Log.js
  LogReview.js
  User.js
  Vehicle.js
node_modules
routes
  analytical_routes.js
  auth_routes.js
  log_review_routes.js
  log_routes.js
  user_routes.js
  vehicle_routes.js
.env
.eslintrc.json
app.js
app.test.js
bcrypt_salt.js
index.js
package-lock.json
package.json
```

- Express used for API server framework
- Application required 4 different models and schemas: Log, Log Review, User and Vehicle
- Seperate middleware for authorisation/authentication and image handling was created as its used extensively throughout different routes

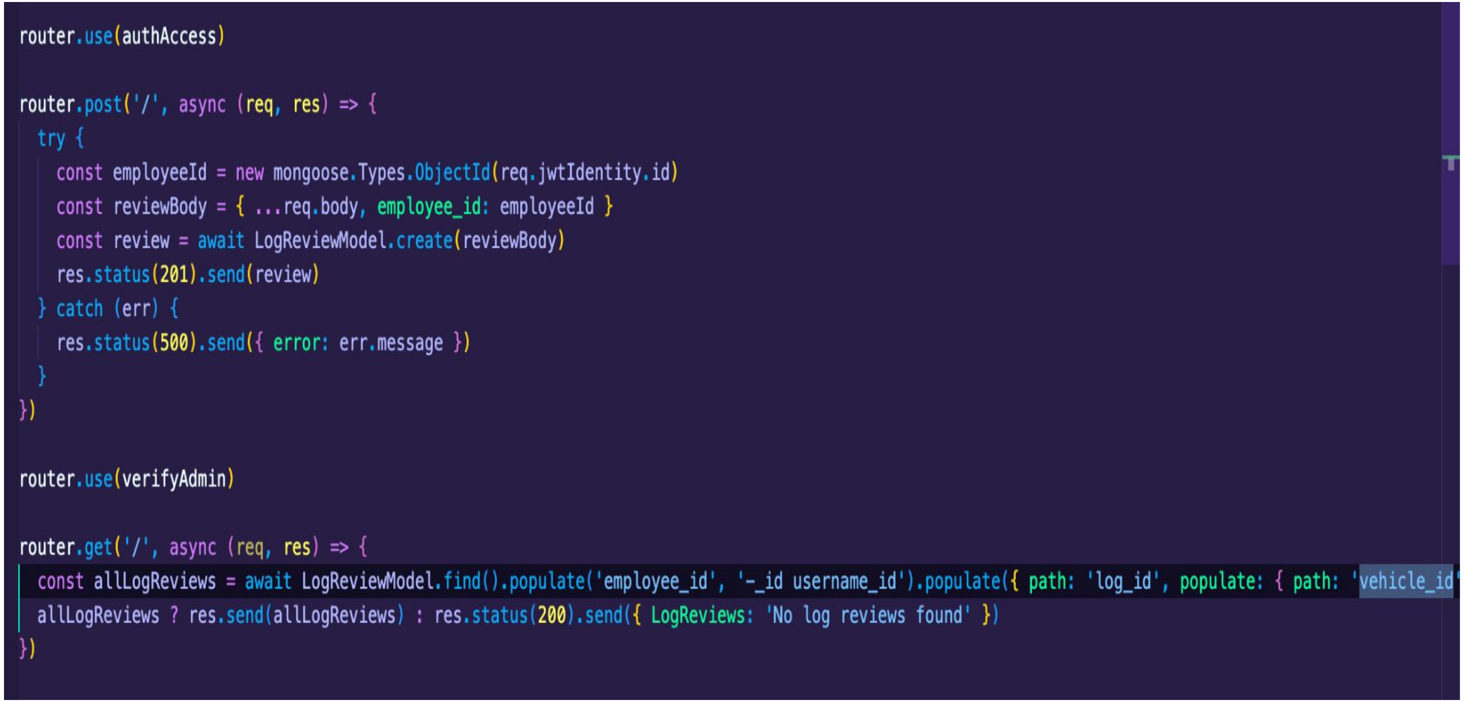
Back end - Middleware



```
// Takes a decoded base64 object, searches if a db exists and sets a cookie in the header with a jwt token
const verifyAuth = async (req, res, next) => {
  try {
    const user = await UserModel.findOne({ username_id: req.userAuthKey.username_id }) // may need to parse to int
    // Need to check if user exists first, otherwise the bcrypt compare throws an error if comparing with a null password
    if (!user) {
      throw new Error('Invalid Username/password')
    }
    const matchPassword = await bcrypt.compare(req.userAuthKey.password, user.password)
    if (!matchPassword) {
      throw new Error('Invalid Username/password')
    } else {
      const jwtUser = { username_id: req.userAuthKey.username_id, isAdmin: user.isAdmin, name: user.name, id: user._id }
      req.isAdmin = user.isAdmin
      req.name = user.name
      // res.setHeader('Set-Cookie', [`accessToken=${jwtGenerate(jwtUser)}; HttpOnly; Max-Age=${24*3600}; SameSite=Secure`])
      res.cookie('accessToken', jwtGenerate(jwtUser), { httpOnly: true, sameSite: 'none', secure: true }) // maxAge was removed so
    }
    next()
  } catch (err) {
    next(err)
  }
}
```

- Middleware mainly provided authentication for logins and authorization to differentiate between an employee and employer
- Example of a authorization middleware that accepts a users login details and sets a JWT in a cookie that is sent back in the response header. This JWT is mainly responsible for providing access to all routes of the API

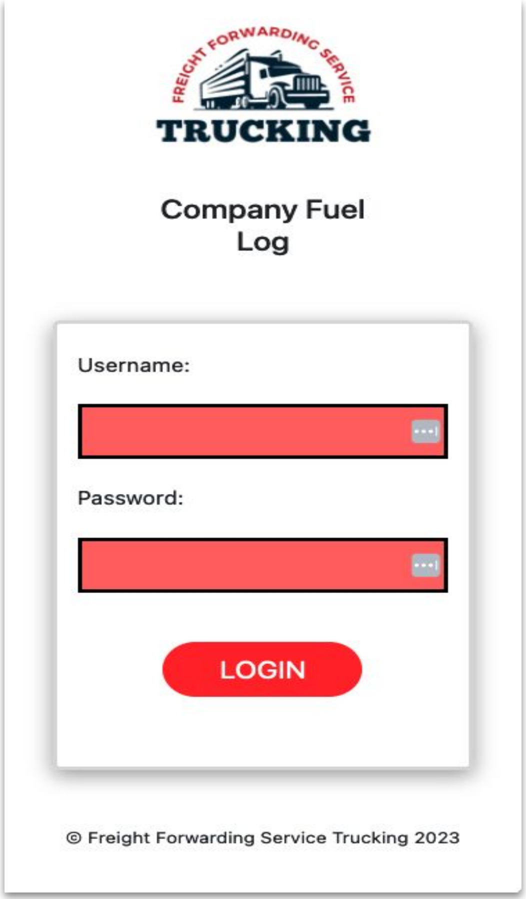
Back end - Routes



```
router.use(authAccess)

router.post('/', async (req, res) => {
  try {
    const employeeId = new mongoose.Types.ObjectId(req.jwtIdentity.id)
    const reviewBody = { ...req.body, employee_id: employeeId }
    const review = await LogReviewModel.create(reviewBody)
    res.status(201).send(review)
  } catch (err) {
    res.status(500).send({ error: err.message })
  }
})

router.use(verifyAdmin)

router.get('/', async (req, res) => {
  const allLogReviews = await LogReviewModel.find().populate('employee_id', '-_id username_id').populate({ path: 'log_id', populate: { path: 'vehicle_id'
  allLogReviews ? res.send(allLogReviews) : res.status(200).send({ LogReviews: 'No log reviews found' })
})
```

- Routes mainly handle validation and processing of incoming requests. Decided that API would handle all data logic so that front end will only just display response data from routes.
- Example show is for getting all Log reviews and post a log review. The middleware was used where the posting of a log review could be done by an employee but retrieving log reviews could only be done by employers

Production Version - Employee section

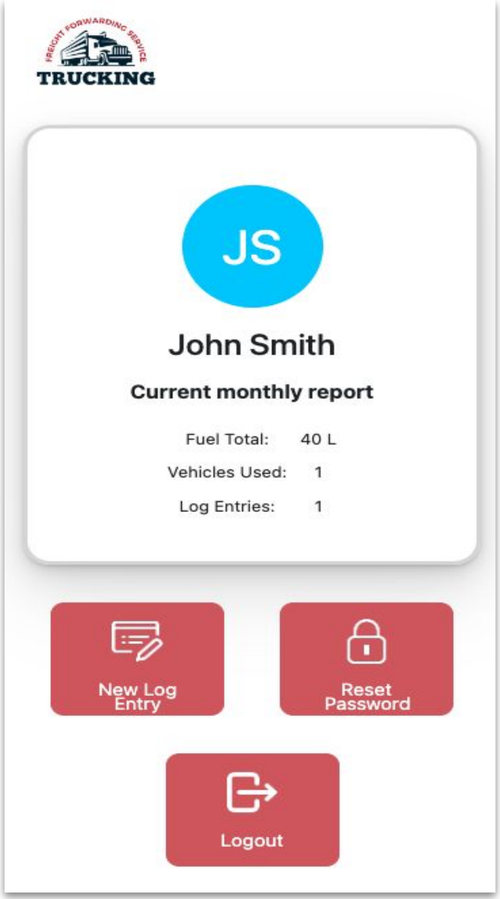


Company Fuel Log

Username:

Password:

© Freight Forwarding Service Trucking 2023



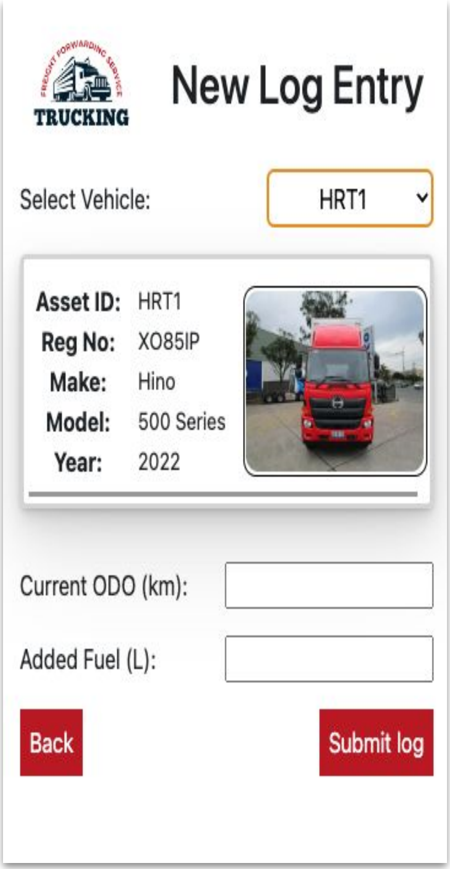
John Smith

Current monthly report

Fuel Total:	40 L
Vehicles Used:	1
Log Entries:	1

- Login as an employee takes to their dashboard with display of monthly report information
- Have option of creating new log, resetting password or logging out of account

Production Version - Employee section



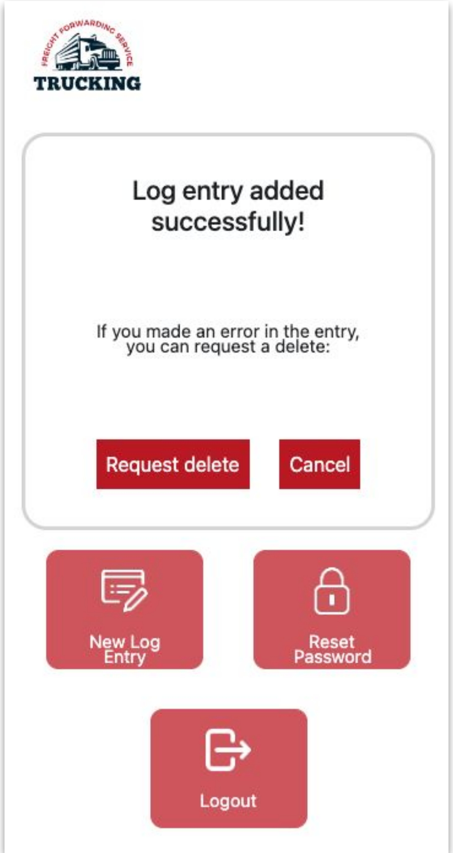
New Log Entry

Select Vehicle: HRT1

Asset ID: HRT1
Reg No: XO85IP
Make: Hino
Model: 500 Series
Year: 2022

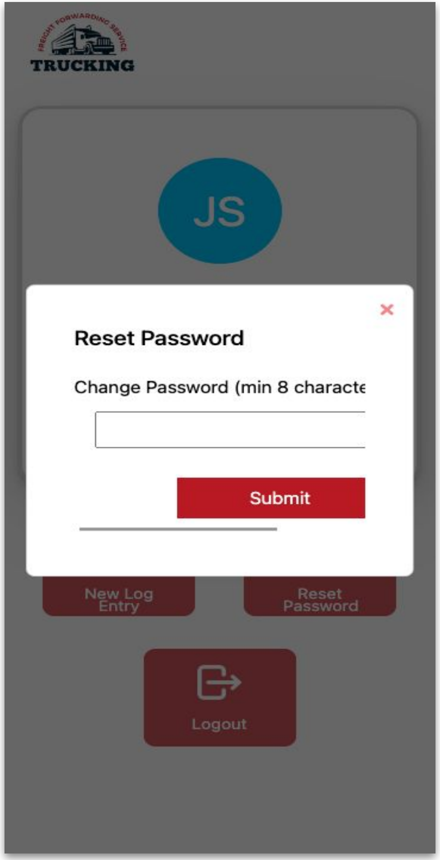
Current ODO (km):

Added Fuel (L):



Log entry added successfully!

If you made an error in the entry, you can request a delete:

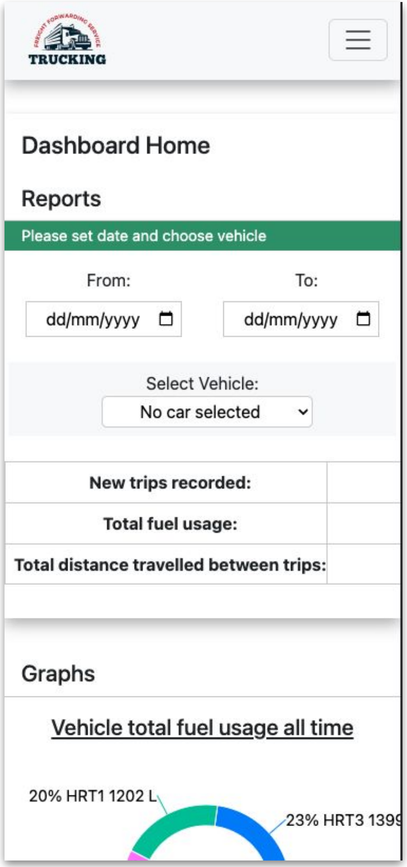


Reset Password

Change Password (min 8 characte

- New log entry can allow vehicle selection and trip information submission
- Successful log reroutes back to dashboard with prompt with log deletion request
- Reset password can change current password, in mobile is a scroll window

Production Version - Employer section



Dashboard Home

Reports

Please set date and choose vehicle

From: dd/mm/yyyy

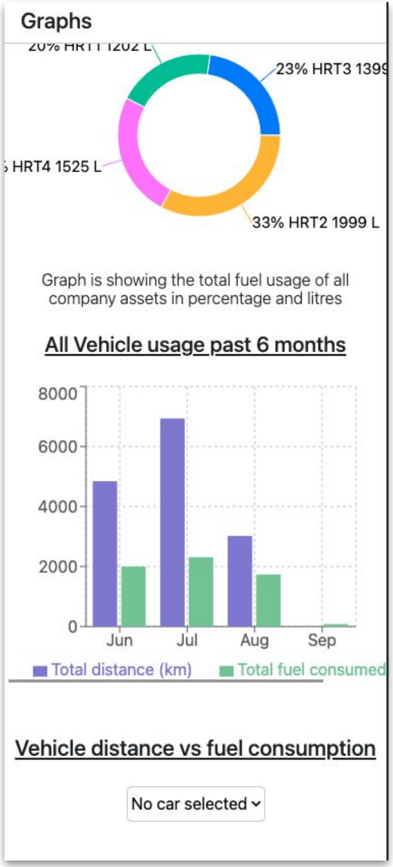
To: dd/mm/yyyy

Select Vehicle: No car selected

New trips recorded:	
Total fuel usage:	
Total distance travelled between trips:	

Graphs

Vehicle total fuel usage all time

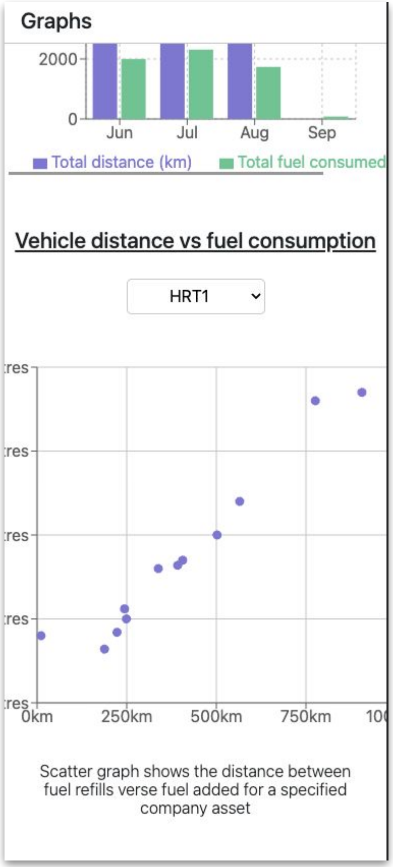

Graphs

Graph is showing the total fuel usage of all company assets in percentage and litres

All Vehicle usage past 6 months

Vehicle distance vs fuel consumption

No car selected


Graphs

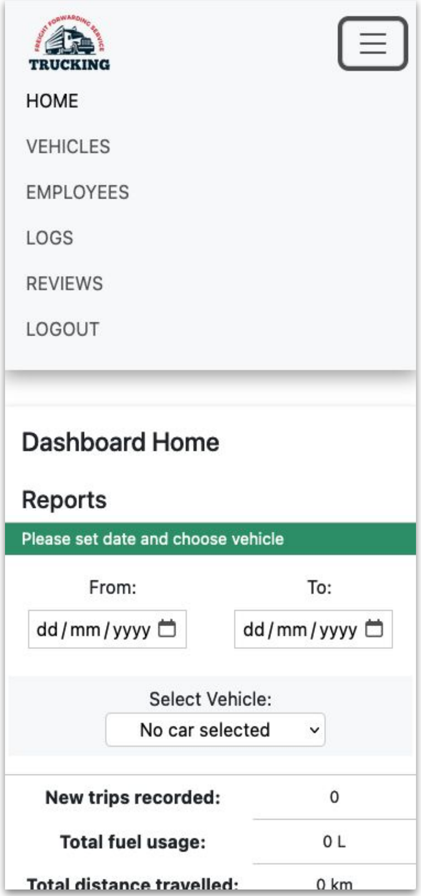
Vehicle distance vs fuel consumption

HRT1

Scatter graph shows the distance between fuel refills verse fuel added for a specified company asset

- Logging in as employer takes to separate dashboard to employee
- Dashboard has date range and vehicle selection that will display distance and fuel information
- 3 separate graphs that convey fuel and distance information of company assets, all set to scroll windows for mobile so graphs can be kept large and still be usable

Production Version - Employer section



HOME

VEHICLES

EMPLOYEES

LOGS

REVIEWS

LOGOUT

Dashboard Home

Reports

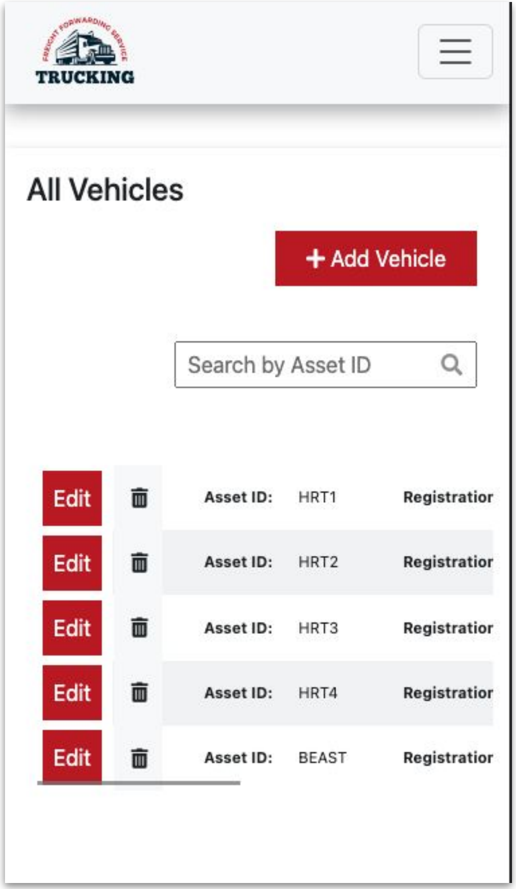
Please set date and choose vehicle

From: dd/mm/yyyy

To: dd/mm/yyyy

Select Vehicle: No car selected

New trips recorded:	0
Total fuel usage:	0 L
Total distance travelled:	0 km

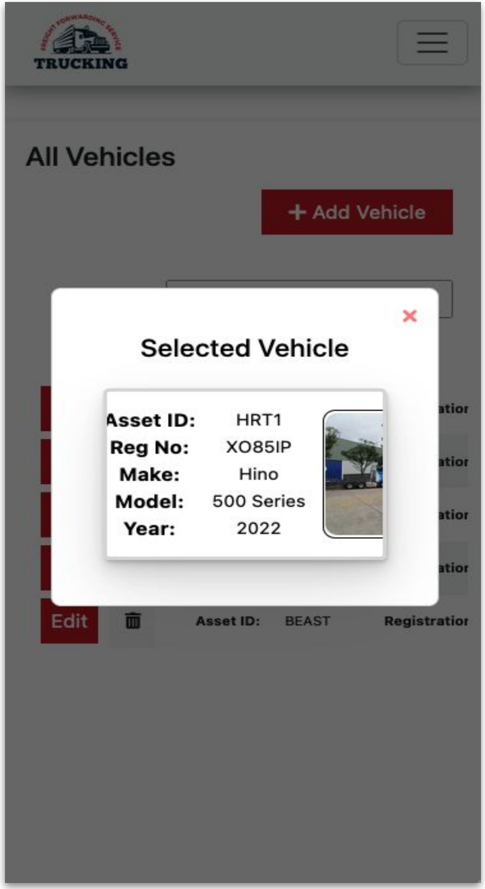


All Vehicles

+ Add Vehicle

Search by Asset ID

Edit	🗑	Asset ID:	HRT1	Registratior
Edit	🗑	Asset ID:	HRT2	Registratior
Edit	🗑	Asset ID:	HRT3	Registratior
Edit	🗑	Asset ID:	HRT4	Registratior
Edit	🗑	Asset ID:	BEAST	Registratior



All Vehicles

+ Add Vehicle

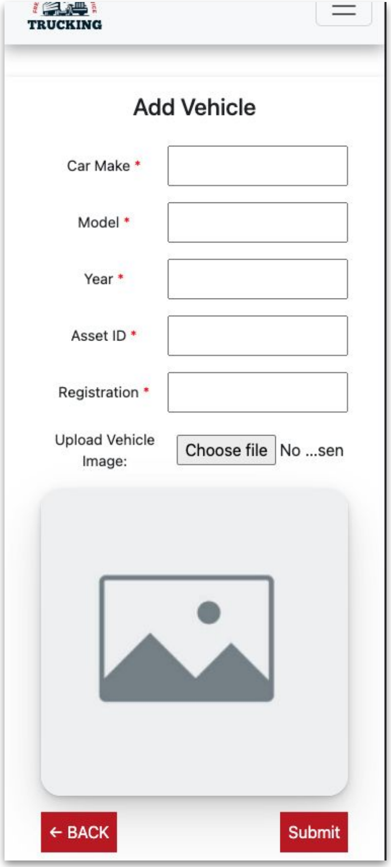
Selected Vehicle

Asset ID:	HRT1
Reg No:	XO85IP
Make:	Hino
Model:	500 Series
Year:	2022

Edit 🗑 Asset ID: BEAST Registratior

- Navbar at top of dashboard drops a menu down to navigate to other sections
- Vehicles tab displays list of vehicles. Can also add, modify or remove company assets.
- Clicking on vehicle in list displays information and image, set to scroll on mobile so image can be set large while having eligible text

Production Version - Employer section



Add Vehicle

Car Make *

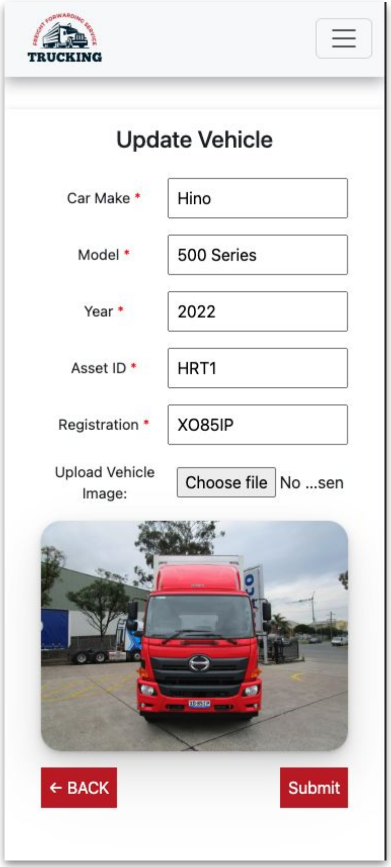
Model *

Year *

Asset ID *

Registration *

Upload Vehicle Image: Choose file No ...sen



Update Vehicle

Car Make * Hino

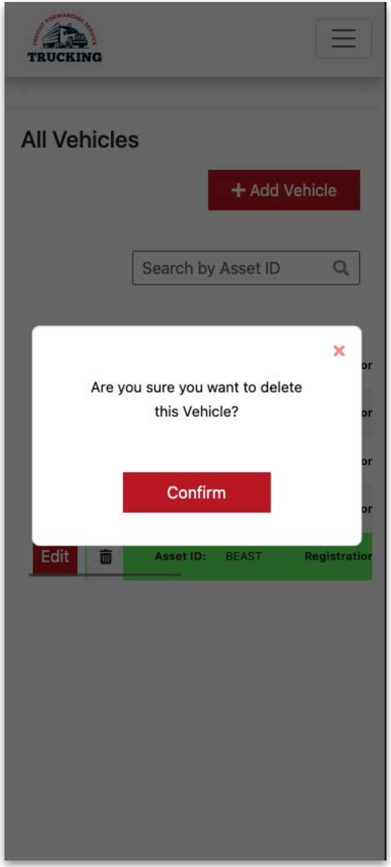
Model * 500 Series

Year * 2022

Asset ID * HRT1

Registration * XO85IP

Upload Vehicle Image: Choose file No ...sen



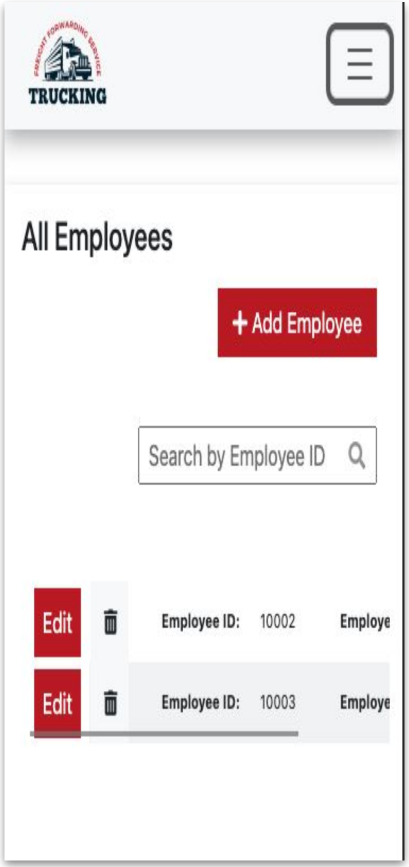
All Vehicles

Are you sure you want to delete this Vehicle?

Asset ID: BEAST Registratio

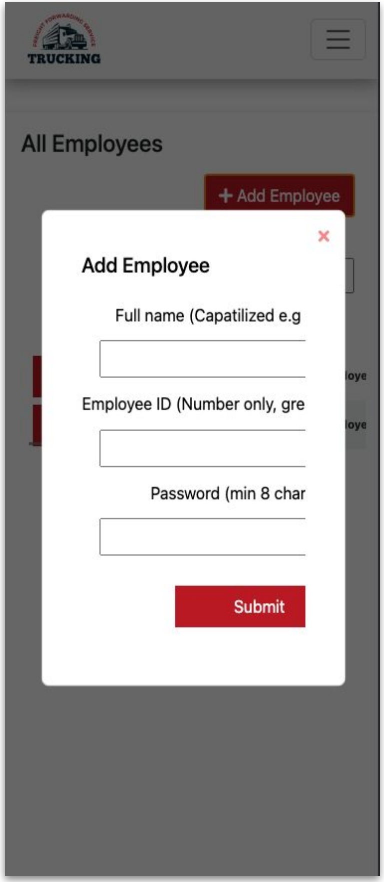
- From vehicles can add vehicles by inputting vehicle information and image
- Updating vehicle will popular input boxes with current information
- Deleting a vehicle will prompt for confirmation

Production Version - Employer section



All Employees

Edit	Employee ID:	10002	Employe
Edit	Employee ID:	10003	Employe



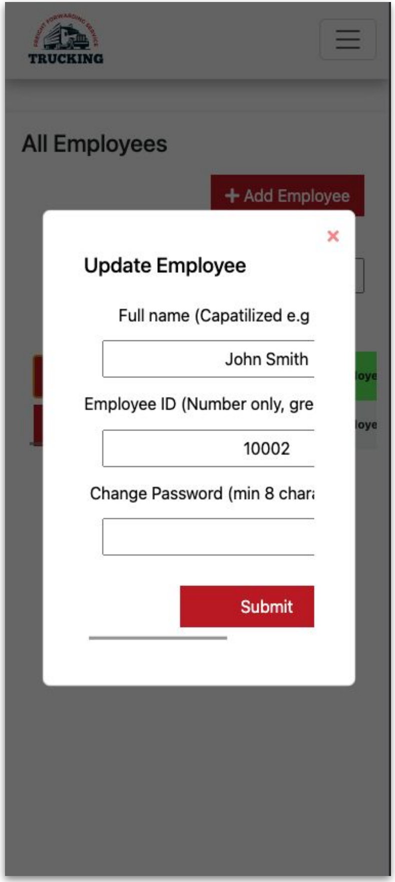
All Employees

Add Employee

Full name (Capatilized e.g

Employee ID (Number only, gre

Password (min 8 char



All Employees

Update Employee

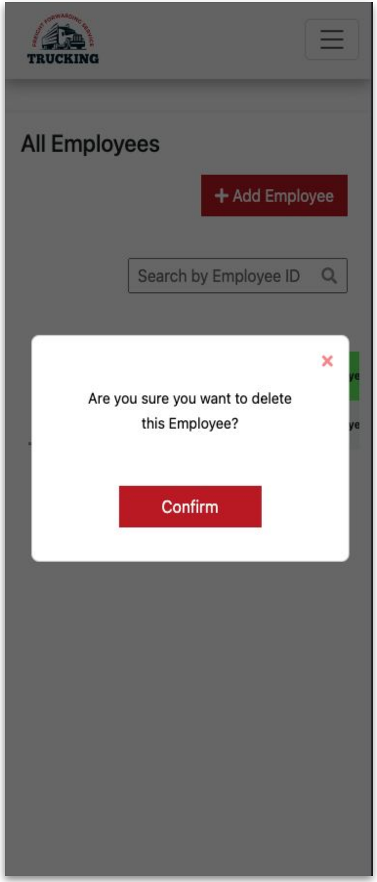
Full name (Capatilized e.g

John Smith

Employee ID (Number only, gre

10002

Change Password (min 8 chara

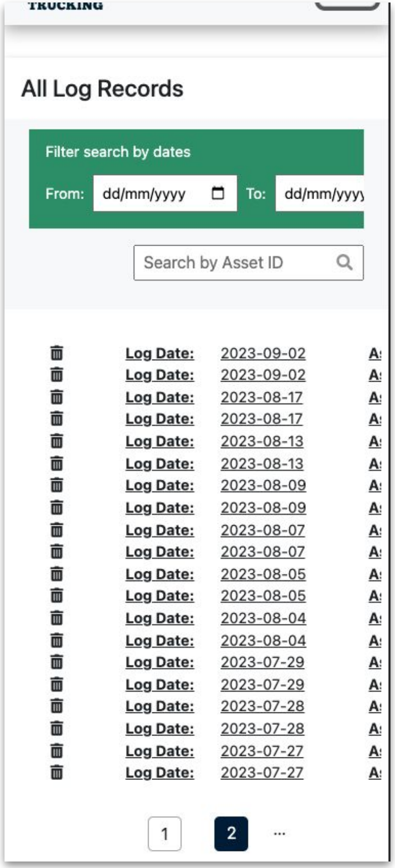


All Employees

Are you sure you want to delete this Employee?

- Employees tab can add, update, delete employees
- Adding employee pops up prompt to input information
- Updating employee populates boxes with current info
- Delete employee asks for confirmation
- All set to have scroll to provide large text and many fields for mobile

Production Version - Employer section



TRUCKING

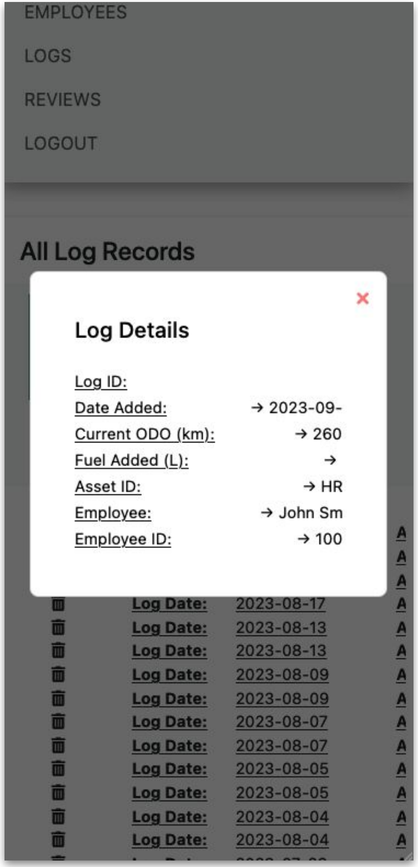
All Log Records

Filter search by dates

From: dd/mm/yyyy To: dd/mm/yyyy

Search by Asset ID

Log Date: 2023-09-02 A:
Log Date: 2023-09-02 A:
Log Date: 2023-08-17 A:
Log Date: 2023-08-17 A:
Log Date: 2023-08-13 A:
Log Date: 2023-08-13 A:
Log Date: 2023-08-09 A:
Log Date: 2023-08-09 A:
Log Date: 2023-08-07 A:
Log Date: 2023-08-07 A:
Log Date: 2023-08-05 A:
Log Date: 2023-08-05 A:
Log Date: 2023-08-04 A:
Log Date: 2023-08-04 A:
Log Date: 2023-07-29 A:
Log Date: 2023-07-29 A:
Log Date: 2023-07-28 A:
Log Date: 2023-07-28 A:
Log Date: 2023-07-27 A:
Log Date: 2023-07-27 A:

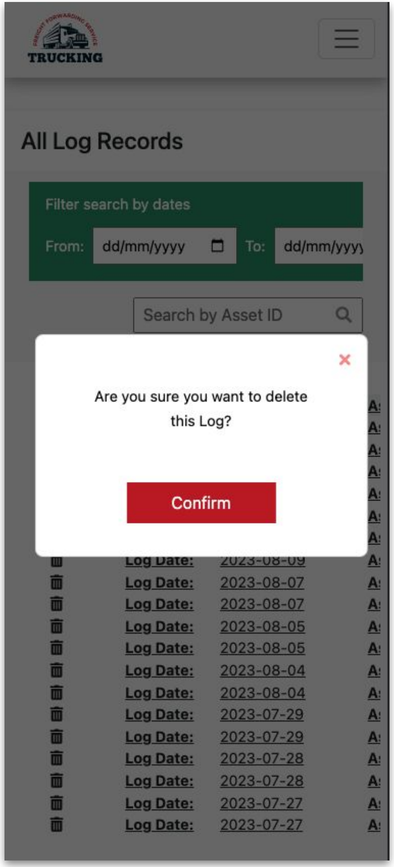


All Log Records

Log Details

Log ID:
Date Added: → 2023-09-
Current ODO (km): → 260
Fuel Added (L): →
Asset ID: → HR
Employee: → John Sm
Employee ID: → 100

Log Date: 2023-08-17 A
Log Date: 2023-08-13 A
Log Date: 2023-08-13 A
Log Date: 2023-08-09 A
Log Date: 2023-08-09 A
Log Date: 2023-08-07 A
Log Date: 2023-08-07 A
Log Date: 2023-08-05 A
Log Date: 2023-08-05 A
Log Date: 2023-08-04 A
Log Date: 2023-08-04 A



TRUCKING

All Log Records

Filter search by dates

From: dd/mm/yyyy To: dd/mm/yyyy

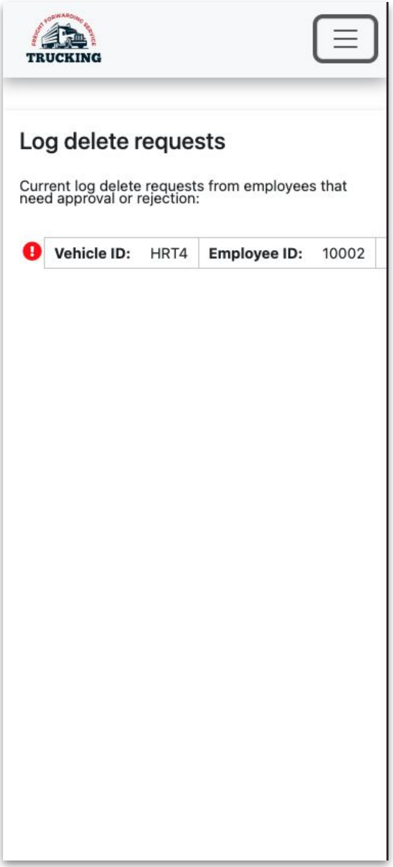
Search by Asset ID

Are you sure you want to delete this Log?

Log Date: 2023-08-09 A:
Log Date: 2023-08-07 A:
Log Date: 2023-08-07 A:
Log Date: 2023-08-05 A:
Log Date: 2023-08-05 A:
Log Date: 2023-08-04 A:
Log Date: 2023-08-04 A:
Log Date: 2023-07-29 A:
Log Date: 2023-07-29 A:
Log Date: 2023-07-28 A:
Log Date: 2023-07-28 A:
Log Date: 2023-07-27 A:
Log Date: 2023-07-27 A:

- Logs display list of trip entries from employees, scrolls in mobile as lot of information to display
- Clicking a log brings up information of trip, asset used and employee. Set to scroll to help with text size for better eligibility
- Deleting a log asks for confirmation

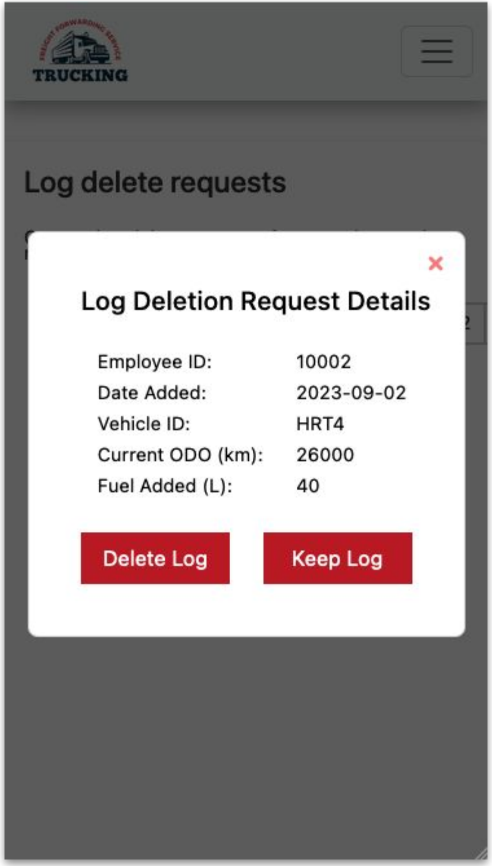
Production Version - Employer section



Log delete requests

Current log delete requests from employees that need approval or rejection:

Vehicle ID:	HRT4	Employee ID:	10002



Log Deletion Request Details

Employee ID:	10002
Date Added:	2023-09-02
Vehicle ID:	HRT4
Current ODO (km):	26000
Fuel Added (L):	40

- Reviews brings up list of log deletion requests from the employees.
- Clicking a request brings up prompt with log trip information. Can either keep log or confirm the log deletion request.

Reflection and Conclusion

- Good learning experience, tests all aspects of web application development from front end with use of react components and styling through to back end with the development of an API server with Node and Express.
- Aesthetics were close to wireframes, in some instances where improved on.
- Some responsiveness is basic verses what is detailed in wireframes. This is mostly due to time constraints which could be improved upon with later versions.
- Overall strived to accomplish the main objective of the project which was to create a simple and intuitive web application that can allow users to add fuel logs and a employer to analyze fuel/distance data and manage employee/vehicle information