PrevWorks

Wellness + Prevention = Productivity
Promoting safety through Prevention

Naila Khalaf, MD, PhD, MPH, M. Arch.
CEO and Founder

Robert Steele, Karalee Mota, Isaac Lee, Cole Howard, Jeff Chen
Development Team

Table of Contents

Project Description

PrevWORKS is a platform designed to foster a safe, healthy and productive work environment for both employees and employers by minimizing workplace risk and promoting the wellbeing and productivity of the workforce. The PrevWORKS platform seeks to evolve the process of determining adverse risks associated with performing certain work activities and maximizing cost savings for employers in response to these risks. PrevWORKS serves employees by streamlining the process of reporting safety concerns, hazardous exposures, and injuries. The PrevWORKS system also aims to efficiently mitigate the challenges employers face when assessing workforce health needs, and determining factors for sustainable wellness and prevention. The PrevWORKS platform consists of the PrevWORKS Web Application and the PrevWORKS API.

The Web Application presents an interactable interface to users (employees and employers), allowing them to view their recorded information and add new data. The Web Application contains two different interfaces, an employee and employer interface, providing different functionality for each type of user. The employee interface allows employees to view and edit their personal information saved within the database. The application allows for wellness information and safety guidelines to be tailored to specific work activities. Injury descriptions and locations of physical injuries are documented through an interactable diagram. In addition to recording their medical history and adding functional capacity reports, employees are able to view their profile and learn about various safety protocols including information on any improvement and change in workplace conditions.

The employer interface in the Web Application allows employers to view and edit their company profile information saved in the database. The employer interface includes pages for the employer to view their company profile, with the option of making changes to reflect updates in the status of employers. Employer interface presents the employer with a summary of the risk assessment of their employees. This also includes an analytics page where the employer can view trends within the employee population and compare their performance to a predictive model based off of industry data. This allows employers to make targeted improvements to their employees by providing any needed information.

The code for this application can be found at: https://github.com/PrevWORKS/PrevWorks

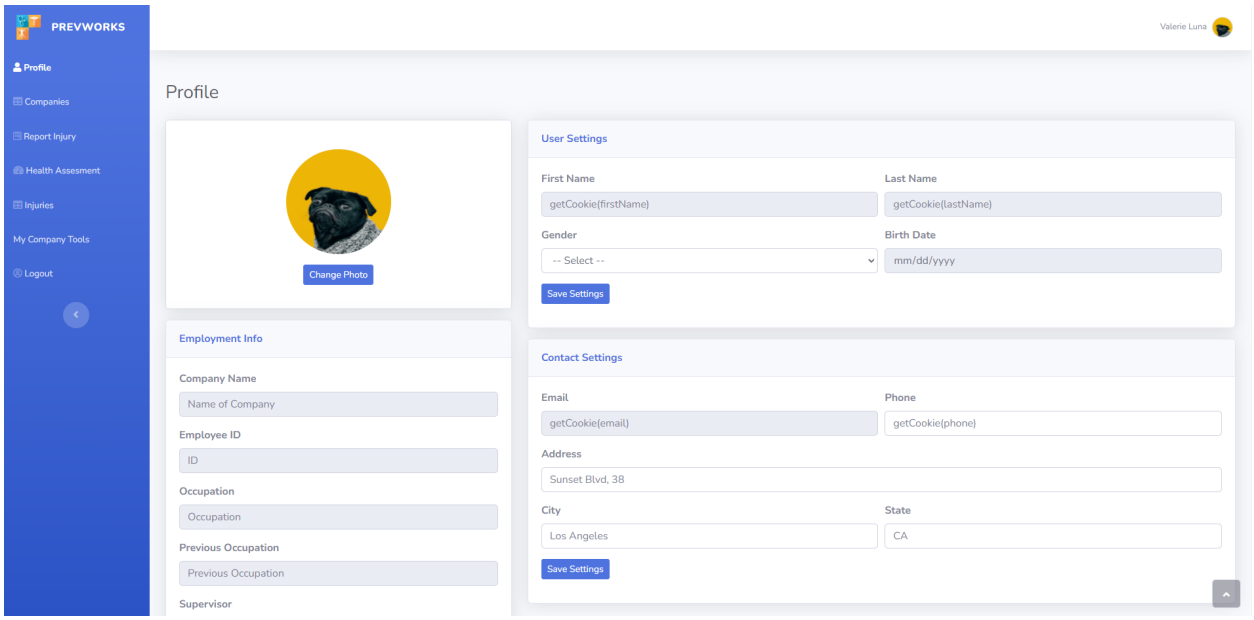
UI Design

Authentication Pages

Employee Pages

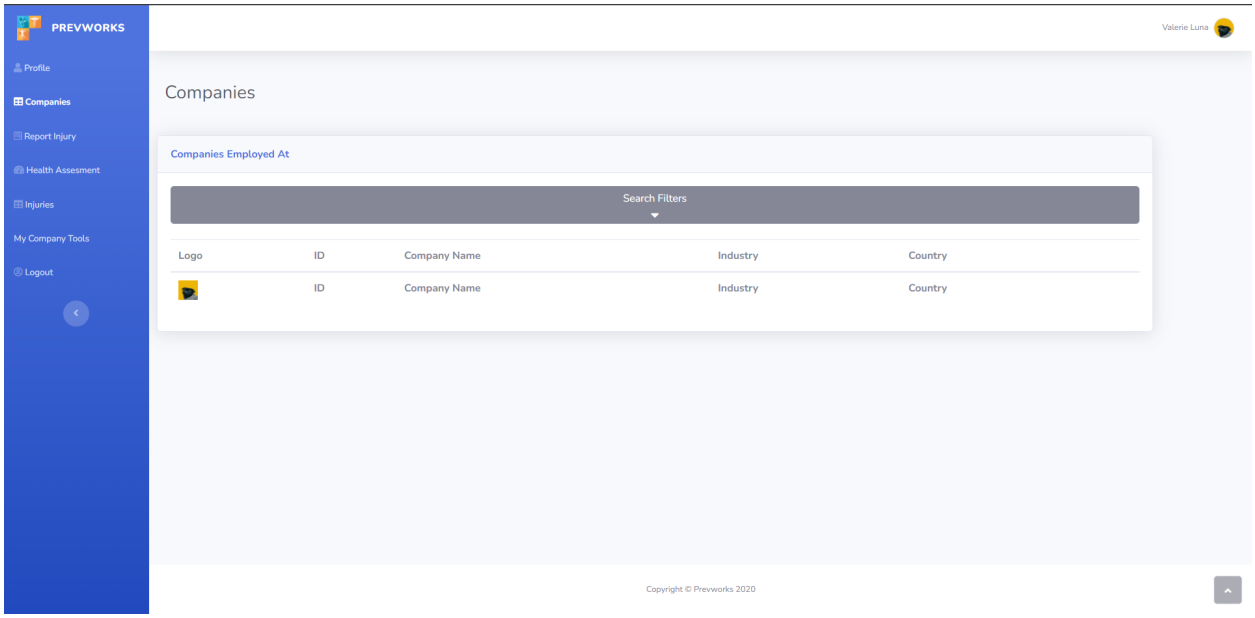
Profile Page

Includes information about the user and forms that allow the user to change information about the user such as address and password.



Companies Page

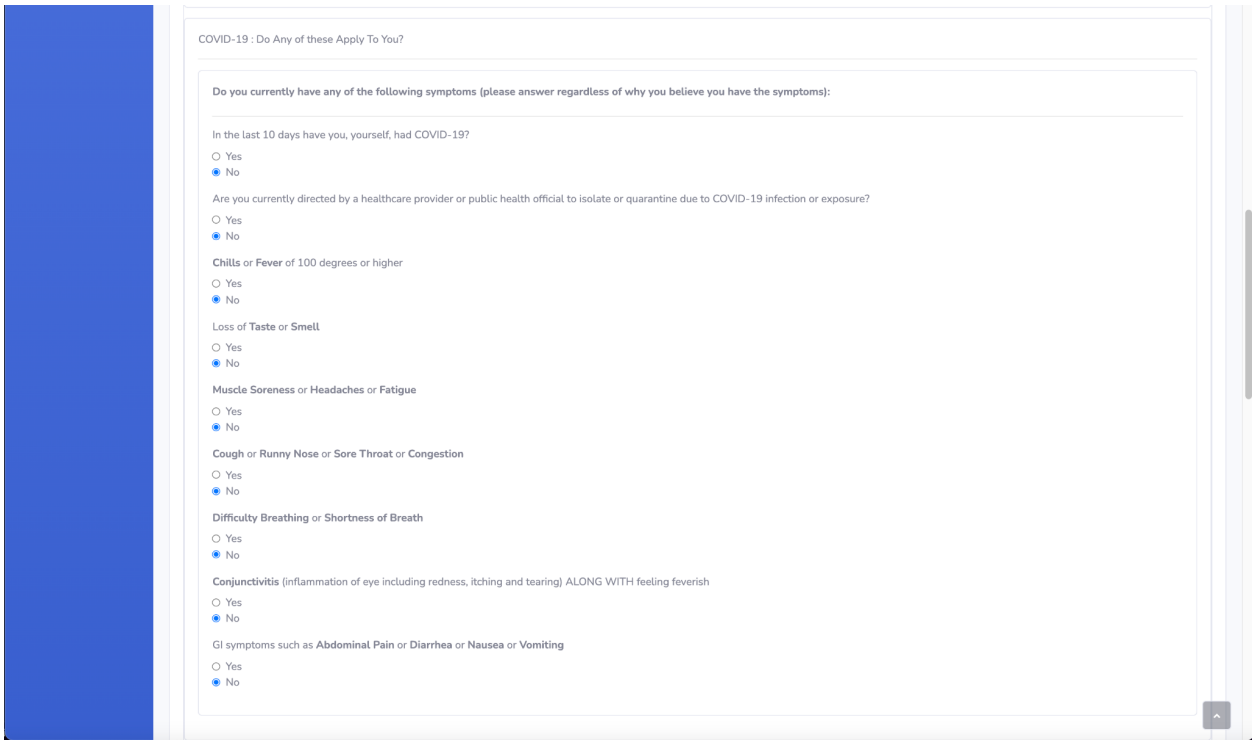
Includes information about the companies that the employee is a part of and allows them to add and remove companies from their list.



Report Injury

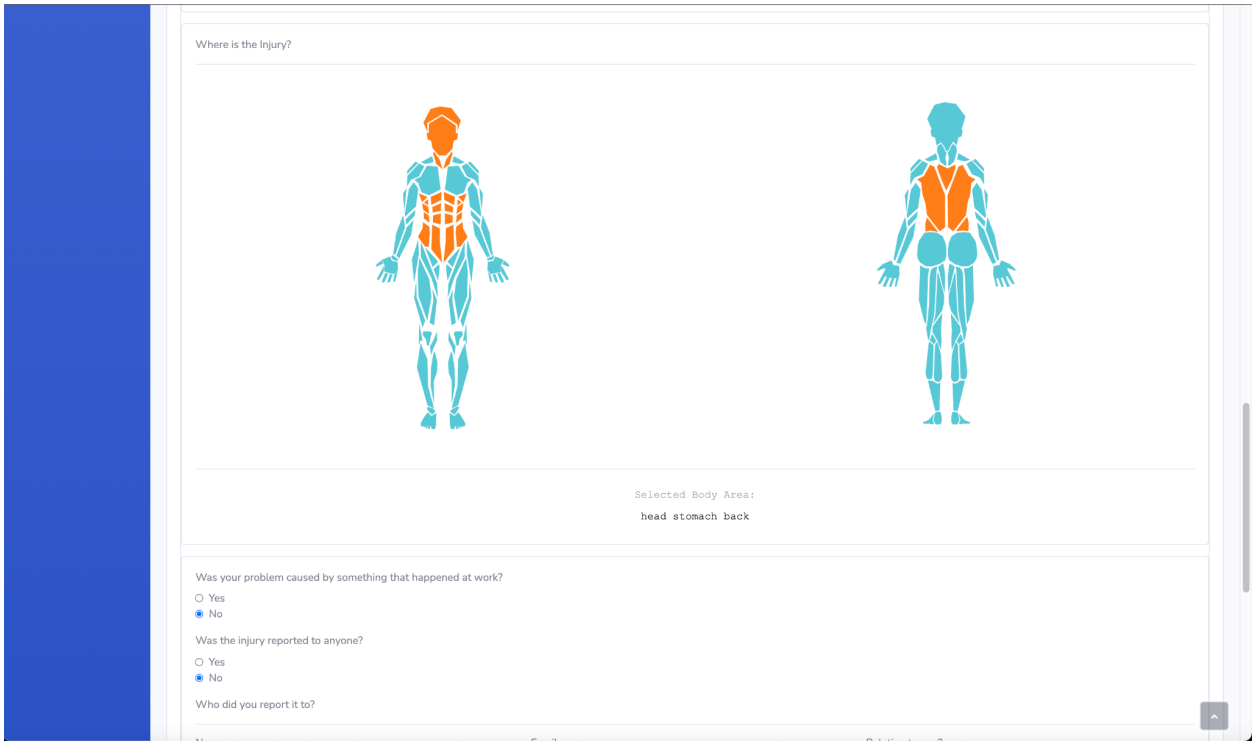
Employees can submit a report of their injuries on the Report Injury page. The page provides a form to report covid-19 symptoms and a body diagram to select injured body parts. Employees can report the injured date and descriptions through the provided text box.

Covid-19 Symptoms Report

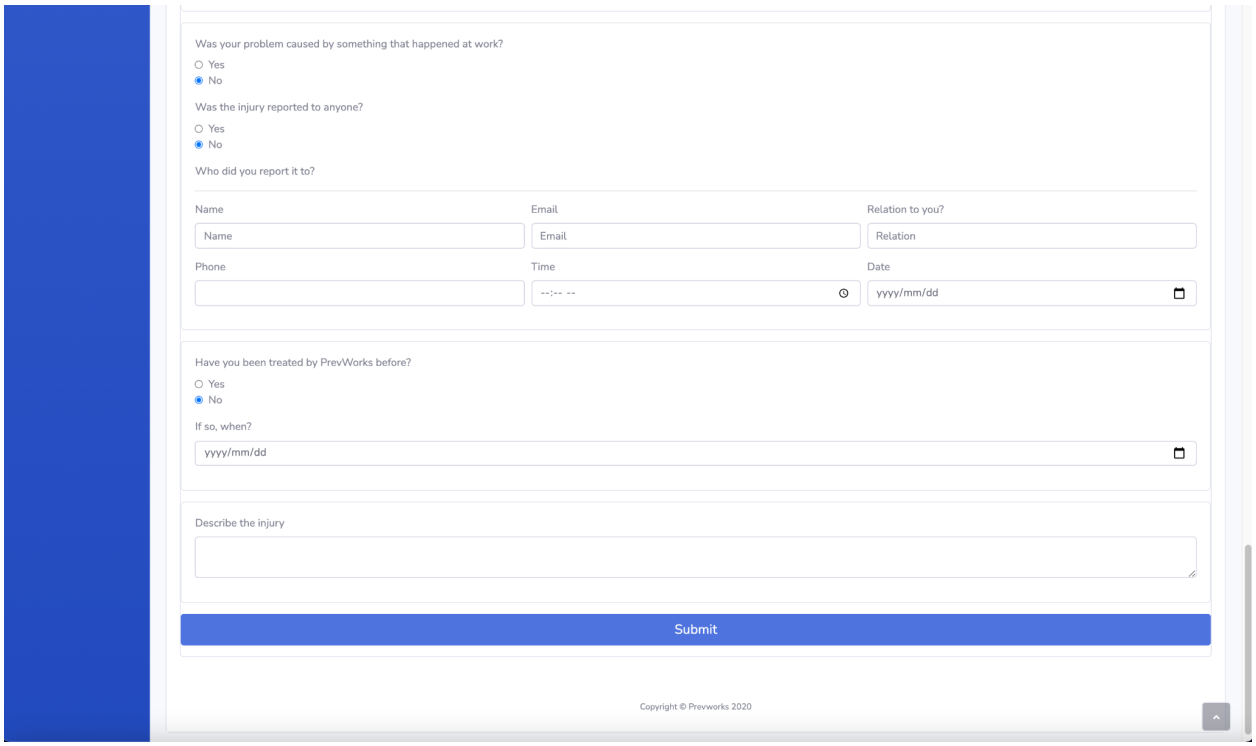


Body Diagram

Employees can report injured body parts by clicking on the body diagram. Multiple body parts can be selected and the selected parts are visually differentiable by color. The color of the selected parts changes to orange. Employees can deselect by clicking on the part again.



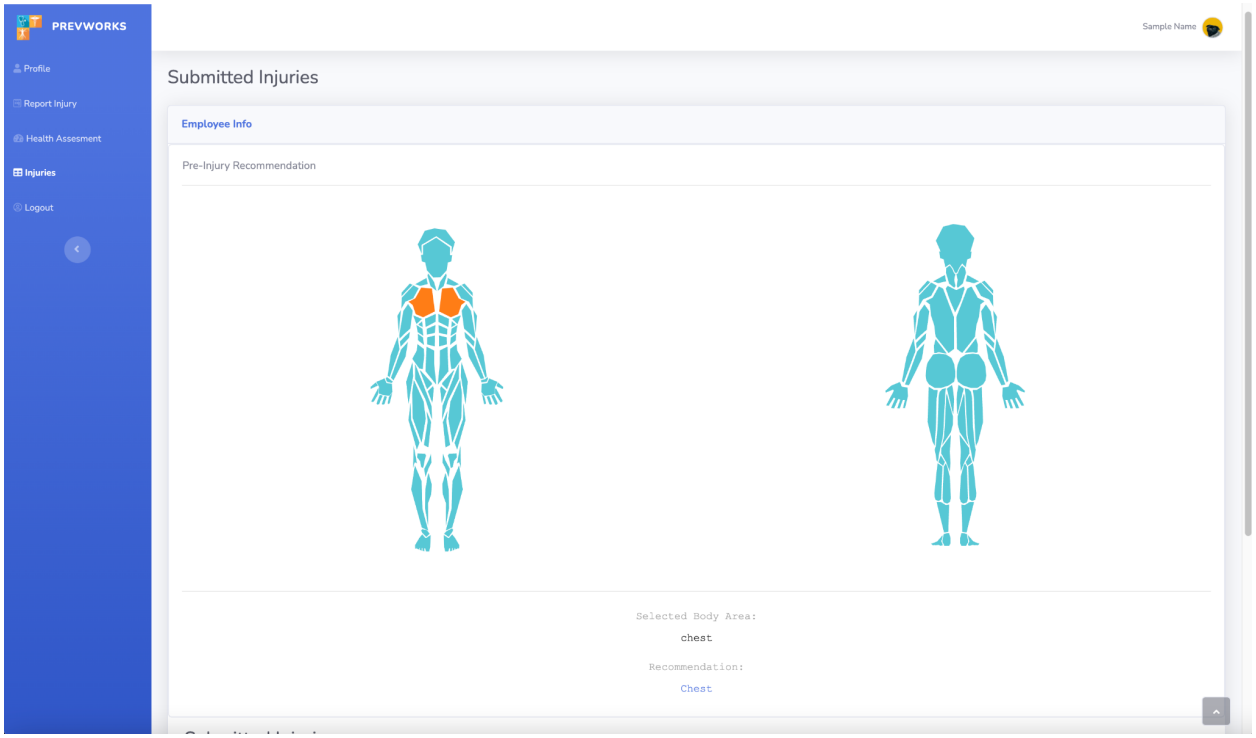
Submit Report



Injuries Page

Before reporting injuries, employees can get information about their aching body parts by selecting the parts in the body diagram. When the part is selected, the color changes to orange, and a link appears at the bottom under Recommendation. Clicking on the link leads the employees to the information page.

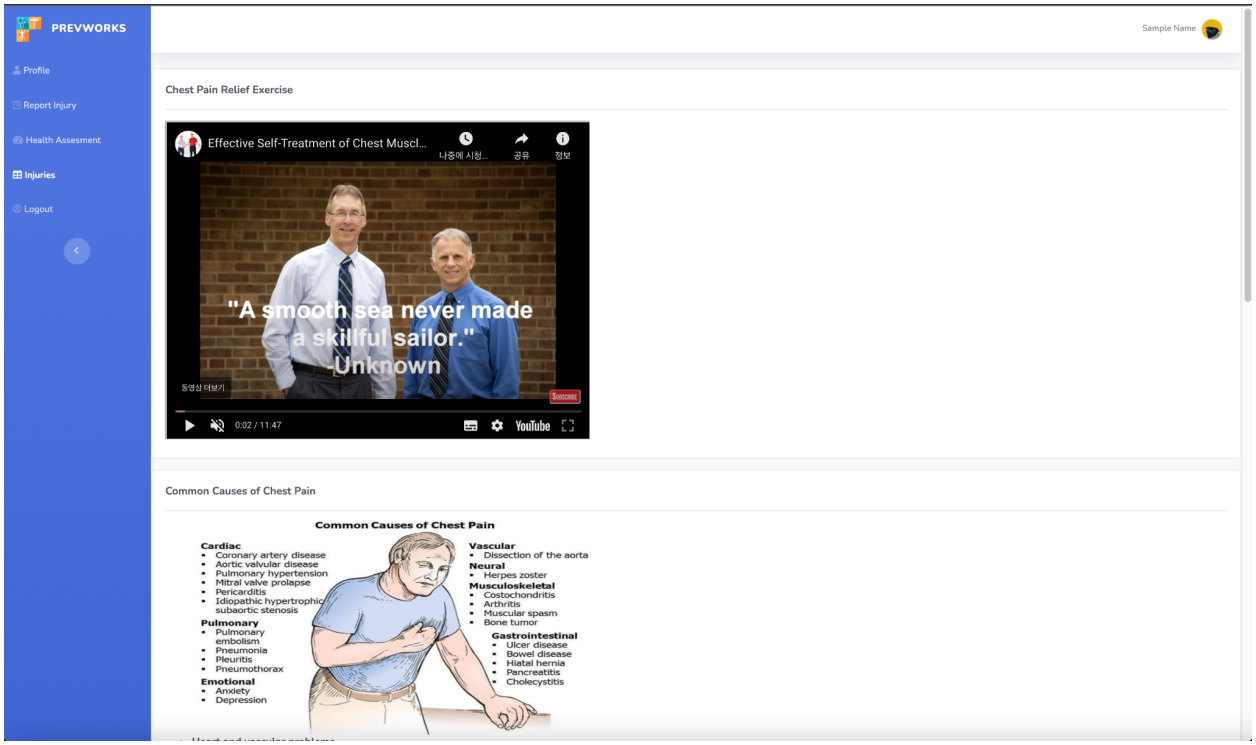
Body Diagram

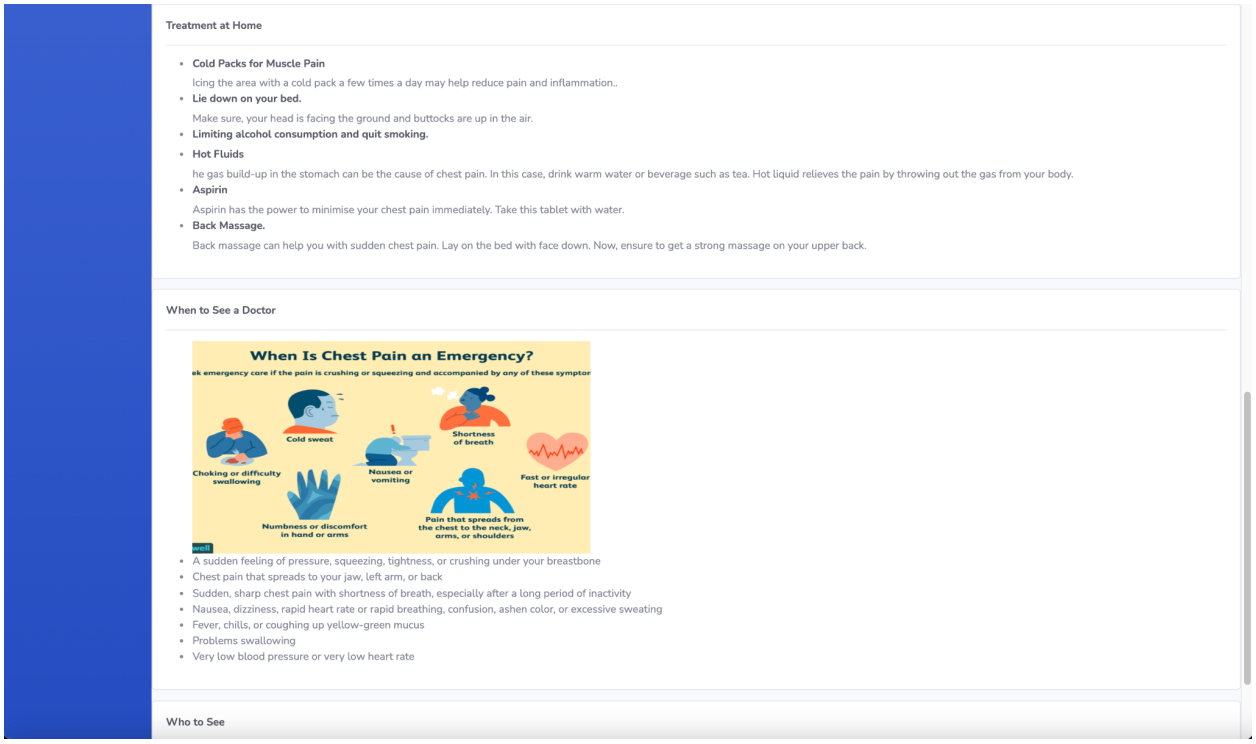


Information Page

For each of the body parts, an information page is provided for the employees to learn about their pain before reporting or visiting a doctor. The information page provides a pain relief exercise video, descriptions of common causes of the pain, possible treatments that can be

done at home, when to see a doctor and which doctor to visit.

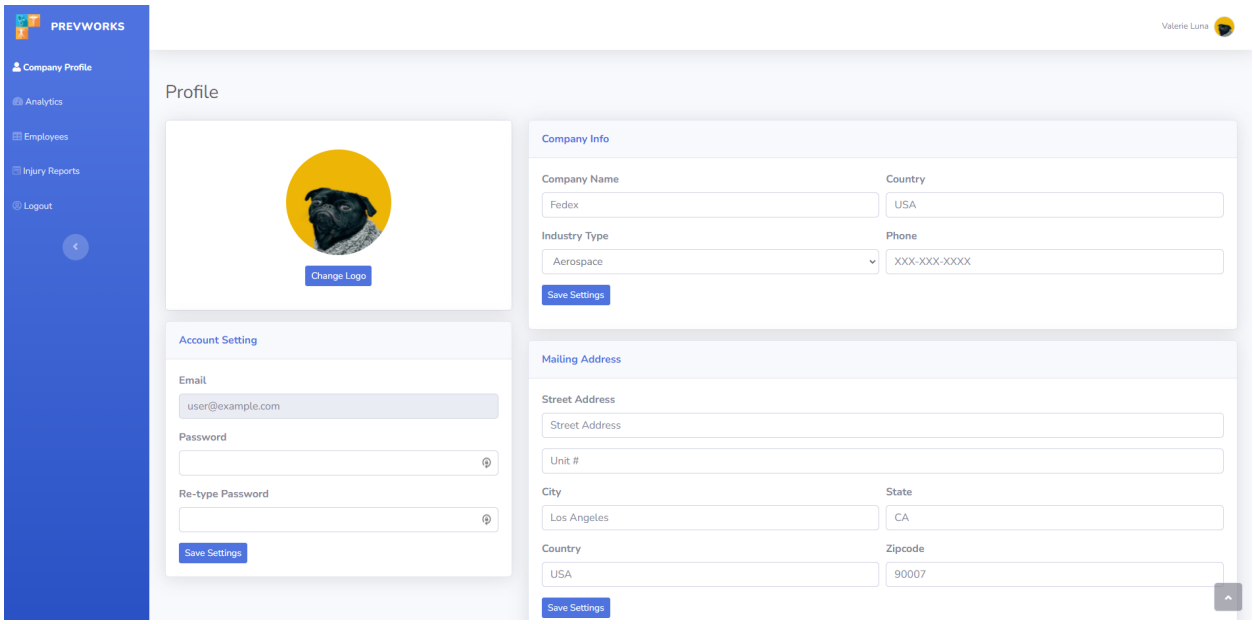




Employer Pages

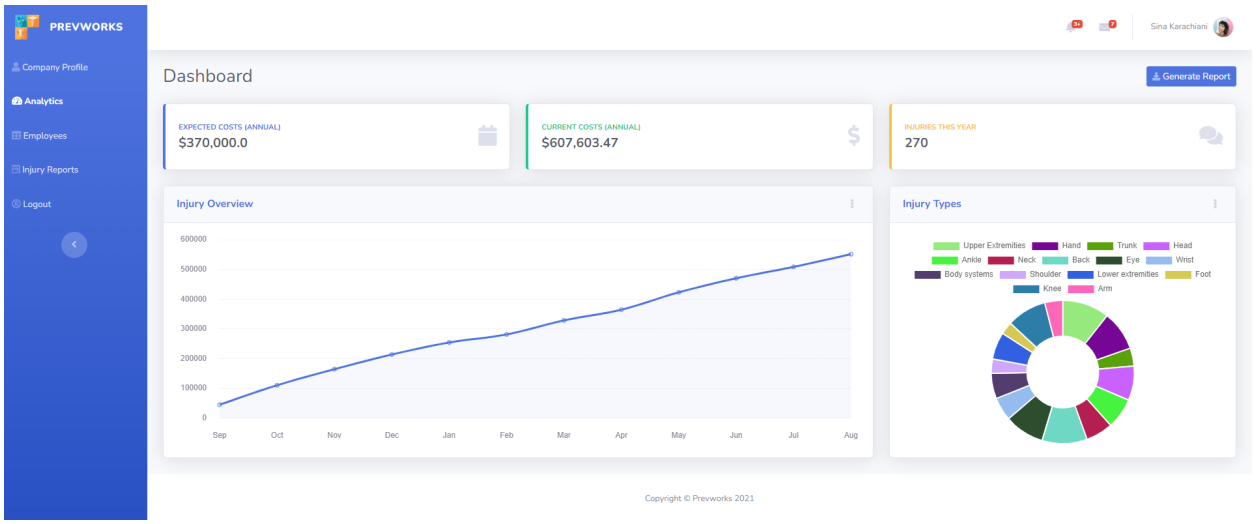
Company Profile Page

Includes information about the company and forms that allow the company to change information about the company such as address and password.



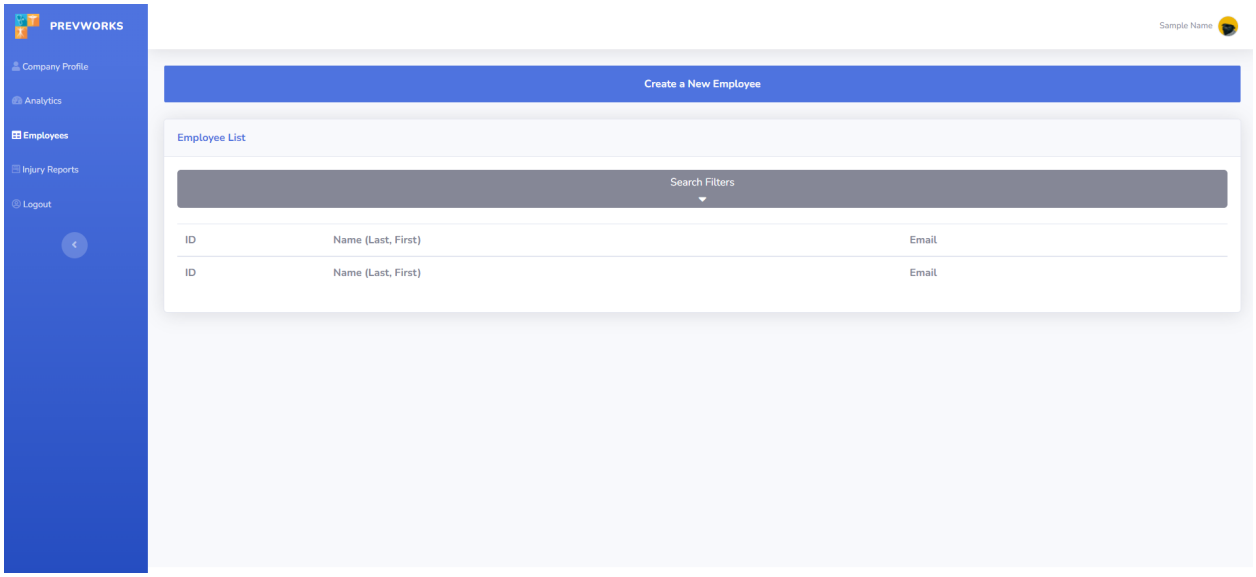
Analytics Page

An employer can view visualizations about their employee population to analyze how different injuries affect the employees. It offers predictions on premiums for your unique population as well as tools to analyze. The components consist of metrics on the top for predicted premiums, actual premiums, and total injuries. Below this there are 2 graphs, one is a line graph for tracking different metrics over time. It also includes a pie chart that has the composition of all the different types of injuries.



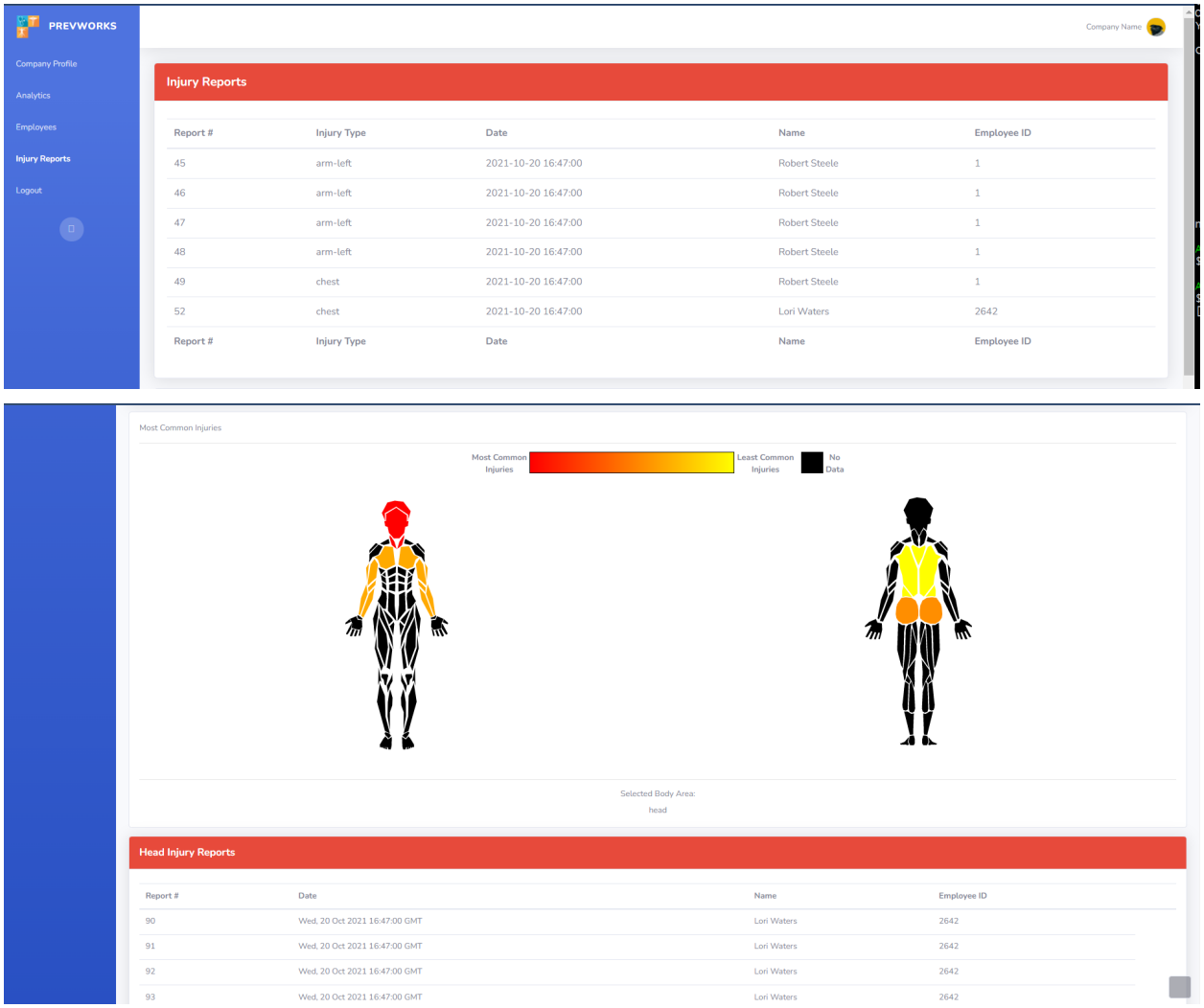
Employees Page

Companies can view their employees on the employee page. This just includes a list of the employees and functionality to add an remove employees from the company.



Injury Reports Page

An employer that owns a company can see a list of all the injuries that have been reported to their company. The report ID number, injury type, date reported, name of employee, and employee ID will be displayed for each injury. Below is a heatmap that shows the most common injuries reported. The body parts colored red are the most common, and yellow are the least common. When a body part is clicked a table pops up that shows all of the injuries of that type that have been reported.



Database Structure

PrevWorks stores data for users, companies, and national injuries. These are stored in a MySQL database composed of 13 different tables. These tables are described below.

Tables

User

Column Name	Data Type	Description
userId	int	Primary key of this table. Each user has a unique userId to join on other tables

email	varchar(100)	Email of the user
nameFirst	varchar(45)	First name of the user
nameLast	varchar(45)	Last name of the user
password	varchar(102)	Encrypted password for the user
occupationId	int	Foreign key to the occupations table

Company

Column Name	Data Type	Description
companyid	int	Primary key of the company Table
companyName	varchar(100)	Name of the company
phone	int	Phone number of the company
streetAddress	varchar(200)	streetAddress for the company
city	varchar(100)	City of address
state	varchar(50)	State of address
zip	int	Zip code of address
password	varchar(102)	Encrypted password for company
loginName	varchar(100)	Name used for logging in for company

Injury

Column Name	Data Type	Description
injuryId	int	Primary key of this table

userId	int	Foreign key to user table
dateOccured	datetime2	Date and time that the injury occured
injuryType	varchar(45)	Location of the injury
at_work	bit(1)	Injury occurred at work
reported	bit(1)	Injury was to anyone
supervisor	varchar(45)	Supervisor injury was reported to
supervisor_email	varchar(100)	Email of supervisor
supervisor_relation	varchar(45)	Relation to supervisor
supervisor_phone	varchar(45)	Phone number of supervisor
supervisor_date	datetime	Date injury was reported to supervisor
reported_before	bit(1)	Have reported an injury to PrevWorks before
description	varchar(1500)	Description of the injury
companyId	int	Id of the company this injury occurred at

Feature

Column Name	Data Type	Description
featureId	int	Primary key of this table
name	nvarchar(100)	Name of the feature

Feature2User

Column Name	Data Type	Description
mappingId	int	Primary key of this mapping table
userId	int	Foreign key to user table

featureId	int	Foreign key to features table
value	nvarchar(100)	Value of the feature

Occupations

Column Name	Data Type	Description
occupationId	int	Primary key of this table
occupationName	varchar(200)	Name of the Occupation

User2Company

Column Name	Data Type	Description
user2companyId	int	Primary key of this mapping table
userId	int	Foreign key to user table
companyId	int	Foreign key to company table
position	int	Foreign key to occupation table

CovidSurvey

Column Name	Data Type	Description
surveyId	int	Primary key of this table
userId	int	Foreign key to user table
date	datetime	Date that the survey was taken
hadcovid	bit(1)	Had covid in the last 10 days
healthcare	bit(1)	Currently a Healthcare worker
fever	bit(1)	Has a fever
loss	bit(1)	Loss of taste or smell

pain	bit(1)	Muscle Soreness, Headache, or fatigue
cough	bit(1)	Cough, Runny nose, Congestion
breath	bit(1)	Trouble breathing
conjunctivitis	bit(1)	Conjunctivitis
gi	bit(1)	Gastrointestinal Issues

CompanyToIndustry

Column Name	Data Type	Description
companyName	int	Name of the company
industry	varchar(400)	Name of the industry
classification	varchar(580)	Classification id for the industry

BodyParts

Column Name	Data Type	Description
bodyId	int	Primary key of this table
name	varchar(45)	Name of the body part

OccupationAgeProbbility

Column Name	Data Type	Description
probabilityId	int	Primary key for this table
ageLower	int	Lower bound on age range
ageUpper	int	Upper bound on age range
probability	float	Probability of injury
occupationId	int	Forgien key to occupation table

OccupationBodyProbbility

Column Name	Data Type	Description
probabilityId	int	Primary key for this table
probability	float	Probability of injury
bodyId	int	Forgien key to body table
occupationId	int	Forgien key to occupation table

OccupationDaysProbbility

Column Name	Data Type	Description
probabilityId	int	Primary key for this table
probability	float	Probability of injury
avgDays	int	Average days away from work
occupationId	int	Forgien key to occupation table

API Structure

The following covers the API documentation for the website. Each header is a component of the path. For example the login header under the auth header would have the route /auth/login. Most of the endpoints do not have an encompassing route. For example register is not under another header so the endpoint is just /register.

The API was built using Flask.

Register

Type: GET or POST
Pre-Req: None
Information Needed: None
Returns: Redirects user to the user register page

Login

Type: GET or POST
Pre-Req: None
Information Needed: None
Returns: Redirects user to the user login page

registerCompany

Type: GET or POST
Pre-Req: None
Information Needed: None
Returns: Redirects user to the Company register page

loginCompany

Type: GET or POST
Pre-Req: None
Information Needed: None
Returns: Redirects user to the Company login page

Profile

Type: GET or POST
Pre-Req: Logged in as User
Information Needed: None
Returns: Redirects user to their profile page

reportInjury

Type: GET or POST
Pre-Req: Logged in as User
Information Needed: None
Returns: Redirects user the report injury page

Companies

Type: GET or POST
Pre-Req: Logged in as User
Information Needed: None
Returns: Redirects user to the companies page

Table

Type: GET or POST
Pre-Reqs: Logged in as user
Information Needed: None
Returns: Redirects user to injuries page with has a table of all injuries

Shoulder

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the shoulder page

Head

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the head page

Chest

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the chest page

Ankle

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the ankle page

Knee

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the knee page

Leg

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the leg page

Neck

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the Neck page

Stomach

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the stomach page

Arm

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the arm page

Elbow

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the elbow page

Wrist

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the wrist page

Hand

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the hand page

Foot

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the foot page

Hip

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the hip page

Back

Type: GET or POST
Pre-Req: Logged in as user
Information Needed: None
Returns: Redirects user to the back page

companyProfile

Type: GET or POST
Pre-Req: Logged in as Company
Information Needed: None
Returns: Redirects user to the company profile page

Analytics

Type: GET or POST
Pre-Req: Logged in as Company
Information Needed: None
Returns: Redirects user to the analytics page

Employees

Type: GET or POST
Pre-Req: Logged in as Company
Information Needed: None
Returns: redirects user the the employees page

Injuries

Type: GET or POST
Pre-Req: Logged in as Company
Information Needed: None
Returns: redirects user to the injuries page

Auth

Register

Type: POST
Pre-Req: None
Information Needed:
- Email
- Password
- First_name
- Last_name

Optional Information:
- StreetAddress

City
Zip
Country
State
phone

Returns: Registers a user and redirects them to the profile page on success

Login

Type: POST
Pre-Req: None
Information Needed:
Email
Password
Returns: Logs a user in and redirects them to the profile page on success

Logout

Type: GET or POST
Pre-Req: None
Information Needed: None
Returns: Clears the session and returns the user to the user login page

registerCompany

Type: POST
Pre-Req: None
Information Needed:
companyName
Email
Password
Industry
employeeClassification
Optional Information
StreetAddress
City
Zip
Country
State
phone

Returns: registers user and redirects them to the company profile page on success

loginCompany

Type: POST
Pre-Req: None
Information Needed:

Login Name
Password

Returns: logs in user and redirects them to the company profile page on success

Forms

sendReport

Type: POST
Pre-Req: Logged in as user
Information Needed:

problemType
(
Date_input
Time_input
Injury_type
At_work
Reported
Reported_before
)
Or
(
Hadcovid
Healthcare
Loss
Pain
Fever
Cough
Breath
Conjunctivitis
gi
)

Optional information:

Supervisor_name
Supervisor_email
Supervisor_relation
Supervisor_phone
Supervisor_date
Supervisor_time
Reported_before_date
injury_description

Returns: Logs injury in the database and redirects user back to the profile page

Predictive Modeling

PrevWorks contains 2 predictive models to aid employers in assessing the health of their patient population. The first model focuses on predicting the number of injuries in the employee population. This allows employers to compare their employee population to the expected number of injuries given their unique employee population and industry. The second model focuses on the amount that the company's insurance premiums will increase by. An in depth explanation of these can be found in the following sections.

Injury Prediction

Initially, we tried gathering data from OSHA and building a simple classification model as seen from the following:

1. https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/data-preprocessing.ipynb
2. https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/pre-event-injury-prediction-model.ipynb
3. https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/post-event-injury-prediction-model.ipynb

However, we lacked the resolution of data required to build a predictive model that uses a Machine Learning algorithm to predict the probability of a specified type of injury for each worker in the company -- we would have required a dataset where each row in the denormalized table represented a user's features (age, sex, number of days in work, previous injuries,...) and label (when they got injured / frequency of a specific injury)... We were unable to source for a free dataset that contained such information. Instead, we took the approach of outputting a general expected number of injuries for a company using the following dataset found below.

Source of data

https://www.bls.gov/iif/soii-data.htm/#summary
Section - **Case circumstances and worker characteristics for injuries and illnesses involving days away from work**
Preprocessing Steps
https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/worker_injury_statistics.ipynb

1. Downloaded the following datasets from subsection By **Occupation**
    a. R9. Detailed occupation by selected natures (Number) (XLSX) (HTML)
    b. R10. Detailed occupation by selected parts of body affected (Number) (XLSX) (HTML)
    c. R11. Detailed occupation by selected sources (Number) (XLSX) (HTML)
    d. R12. Detailed occupation by selected events or exposures (Number) (XLSX) (HTML)
    e. R41. Detailed occupation by age of worker (Number) (XLSX) (HTML)
    f. R44. Detailed occupation by industry division (Number) (XLSX) (HTML)

    g. R66. Detailed occupation by number of days away from work (Number) (XLSX) (HTML)
    h. R98. Detailed occupation by selected natures (Rate) (XLSX) (HTML)
    i. R100. Detailed occupation by selected events or exposures (Rate) (XLSX) (HTML)
2. Found sample size for each occupation by dividing R9 by R100 and divided each sample size of the specific occupation by the total number of people surveyed which represents p(Occupation), probability of being in a specified occupation.
3. Divided R10, R11, R12, R41, R44, R66 by the sample size to get their rates which represent p(Injury=True, Body Part | Occupation), p(Injury=True, Source | Occupation), p(Injury=True, Event of Exposure | Occupation), p(Injury=True, Age of worker | Occupation), p(Injury=True, Industry Division | Occupation), p(Injury=True, Days away from work | Occupation) respectively.

Implementation Details

https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/worker_injury_statistics.ipynb
We model the number of injuries in a given company as a random variable drawn from a Poisson Binomial distribution, an extension of a Binomial distribution assuming we have independent trials but with each a different probability of success/injury.

E[Injuries in Company]

We calculate the expected number of injuries as simply the sum of the conditional probabilities we have calculated in the data preprocessing section.

E[(Injuries in Company - E[Injuries in Company])^2]

We calculate the variance of the number of injuries in the company as the dot product of the vector of injury probabilities of each worker with 1 - the probability.

Synthetic data

https://github.com/PrevWORKS/PrevWorks/blob/main/notebooks/populating-database.ipynb
We created random users using helper functions in the previous section and the notebook above to populate the database we fake users for a company.

Workers' Compensation Premium Prediction

Workers' Comp Premium = Employee Classification Rate * Employer Payroll (Per $100) * Experience Mod Rate
Employee classification rate is a dollar amount calculated by NCCI that gives the relative cost of insurance for particular kinds of work. We found these rates here:
https://www.wcirb.com/class-search
The code to scrape this data can be found in scrape_classification_rates.py. Note that it will not scrape the values from the last page. The employer payroll is the total amount of money paid to employees in salary. The experience modification rate is a number that shows how a company

compares to other companies with similar kinds of work based on a number of factors like age of the business, frequency of injuries, and severity of injuries. This number is calculated by the NCCI on a case-by-case basis, so we do not have access to it. Therefore, we replaced it with a ratio of the company's injury incidence rate to the incidence rate of their corresponding industry. The industry incidence rates can be found here:
https://www.bls.gov/iif/oshsum.htm#19Summary_News_Release
All of these factors are combined to calculate the workers' compensation premium, which is how much the employer can expect to pay in insurance for workplace injuries in a given year. The function to compute workers' compensation premium can be found in compensation.py and is called calc_total_comp(). This is what is used to calculate the actual cost shown on the analytics page. To calculate the expected cost shown on the analytics page, we calculated what the industry average cost would be if it was the exact size of the company using our site, meaning the experience modification rate becomes 1. This calculation can be found in compensation.py in the function called calc_industry_premium().

This equation could also be used to calculate an employee's contribution to this total premium by replacing total payroll with employee salary and experience modification rate with the probability of the employee being injured (p(Injury=True, Age of worker | Occupation) as seen in the predictive modeling section). Although we wrote a function in compensation.py called calc_specific_comp() to do this, it was never utilized in the final product.

Deployment

Our implementation of PrevWorks web application is in the form of a flask application. This allows for easy deployment on any server. You can find guides here on how to deploy the application. https://flask.palletsprojects.com/en/2.0.x/deploying/index.html.

For local deployment follow these steps:

1. Set the FLASK_APP and FLASK_ENV variables to flaskr and development respectively. These can be set on windows using "set FLASK_APP=flaskr" and "set FLASK_ENV=development". On linux and mac use "export FLASK_APP=flaskr" and "export FLASK_ENV=development".
2. Run the command "flask run".