Innovation Drive and Technology Expo

User Manual

Chris Ciolek

Nicholas Spencer

Maxx Achtman

Apolonio Cazares

[**Introduction**](#_ley5tu0eqdk) **4**

[Purpose of product](#_13dqg5eu54bo) 4

[Operating environment](#_c2so7n3nd3h9) 4

[General functionality](#_7aj556rvcdmj) 4

[Special features](#_n98tujq3d5m0) 4

[Limitrations](#_riqwtvttimbr) 4

[Documentation conventions](#_5dhnrhok5dnc) 5

[Java](#_e0f4d6vlx3kx) 5

[HTML](#_et36b4vhhs4m) 5

[CSS](#_b0hwfxylmc3u) 5

[JavaScript](#_41hns9lka1ay) 5

[JSX](#_rk2qviigyvjq) 5

[**Installation**](#_8e22vuar4n7j) **5**

[Physical requirements](#_pl7mg4bxmqci) 5

[Copying & backup](#_1ntm044pjoi3) 5

[Software installation](#_v2ijbrbabo7w) 6

[Customizing product](#_18j13jlnityc) 6

[Home Page](#_z61ouz6zaflr) 6

[Footer](#_lsxzkxc63onx) 7

[Information Page](#_1ft1nu91hjdg) 8

[Event Info / FAQ](#_bwg2t31uysi) 8

[Event Map](#_jvmntdbzdkih) 9

[Event Layout](#_ksvw2cdgenxe) 10

[Contact IDTE](#_q531ib3pfeyf) 11

[Digital Waiver Page](#_ijr4fqoosa5b) 11

[**Tutorial**](#_7s6mxsq32ghs) **12**

[Setup the program locally](#_xrriag4jmq4b) 12

[Visual Studio Code](#_p5pg3t1vyi0c) 12

[Microsoft SQL Server Manager](#_ih81ni5fnsjd) 12

[Walkthrough of example](#_tlnidgmel0rq) 14

[Be an attendee and go through the steps](#_n6lm6u57upo4) 14

[Check In the Attendee as an Admin](#_75p8wlz2mhps) 20

[Explanation of example](#_aw7sp2v5n74h) 23

[Using help (on-line & text)](#_p2hhe9bjyekc) 23

[**Detailed Instruction**](#_voovf766c4uc) **24**

[Output from product](#_v28ij7pdncjj) 24

[Input to product](#_bykwq2u6rhqc) 25

[Operation of product](#_l318r6418mqo) 25

[Error handling](#_nxw654ep6sxy) 25

[Specific functions](#_xps7vktst5bz) 25

[How to run](#_pjbm3q3q3cgu) 25

[Input data needed](#_xbeyol5xl6d5) 25

[Output interpretation](#_af3hag1ra6d2) 25

[**Technical Details**](#_oa9wffarnks) **26**

[Principles of operation](#_esvh5ydjbue8) 26

[Advanced features](#_cgzkxjsev4qw) 26

[Modified of product](#_hfq9zmjw8e9f) 26

[Support information](#_diq2dymwu93z) 26

# 

# Introduction

## Purpose of product

This website is used to allow registration for technologies and attendees during Ford’s annual IDTE event. This product can make the event registration and check-in more organized through a site.

## Operating environment

This is a web based application. That means that this site can run on any internet browser and mobile.

## General functionality

Users can register for technologies by entering in technology information and associating with a technology category that the administration would define. Users can also register as one of three attendee types, those being “evaluator”, “supplier”, and “presenter”. Through the attendee registration, an attendee can be associated with setup days, dry run day, and event days. There is also an information tab that would contain Event Information, Event Map, Contact information, and Layout IDTE.

## Special features

There is also administration only view that can be accessed after logging into the site. This login can only be accessible by selecting the admin button next to the copyright at the bottom of our site. Through this page, administration can edit and remove attendees or technologies, set up technology categories, send emails, and export technology and attendee databases. On top of that, administration can create a new event or edit an event that is currently active.

## Limitrations

As mentioned previously, the administration tab can only be viewable after a valid login. So a normal user would not see any of the features mentioned in section 1.4 of this document. So that means security is a big focus on this site and that is obvious with said approach.

## Documentation conventions

There are different types of files that have been created to make this project possible. Below are the different types of files that are needed for this project to run.

### Java

Java files are built as the backbone of the project. Any functionality that is generated is run through these Java files. These files would be talking to the databases that are built through SQLLite.

### HTML

These files are created to generate the web pages that an end user can see.

### CSS

This language is used to set up the style of the site. Page layouts, colors, and fonts are all set up with CSS.

### JavaScript

This language is used to create dynamic web pages. This file type in our project is used to link the HTML page with a backend functionality.

### JSX

This file type is used to communicate with React components for our site. In these files, we are building the site itself using these file types.

# Installation

## Physical requirements

* A PC is required with valid internet and web browser to access the site
* Our site can also be ran on mobile if need be, but this is mainly used for an internet browser

## Copying & backup

Using GitHub as our repository, we can have this be accessible to the client for the code if any errors may occur during any changes. The databases are currently held locally, but once they launch onto the server, we are passing it over to Ford IT to handle the backup of the databases.

## Software installation

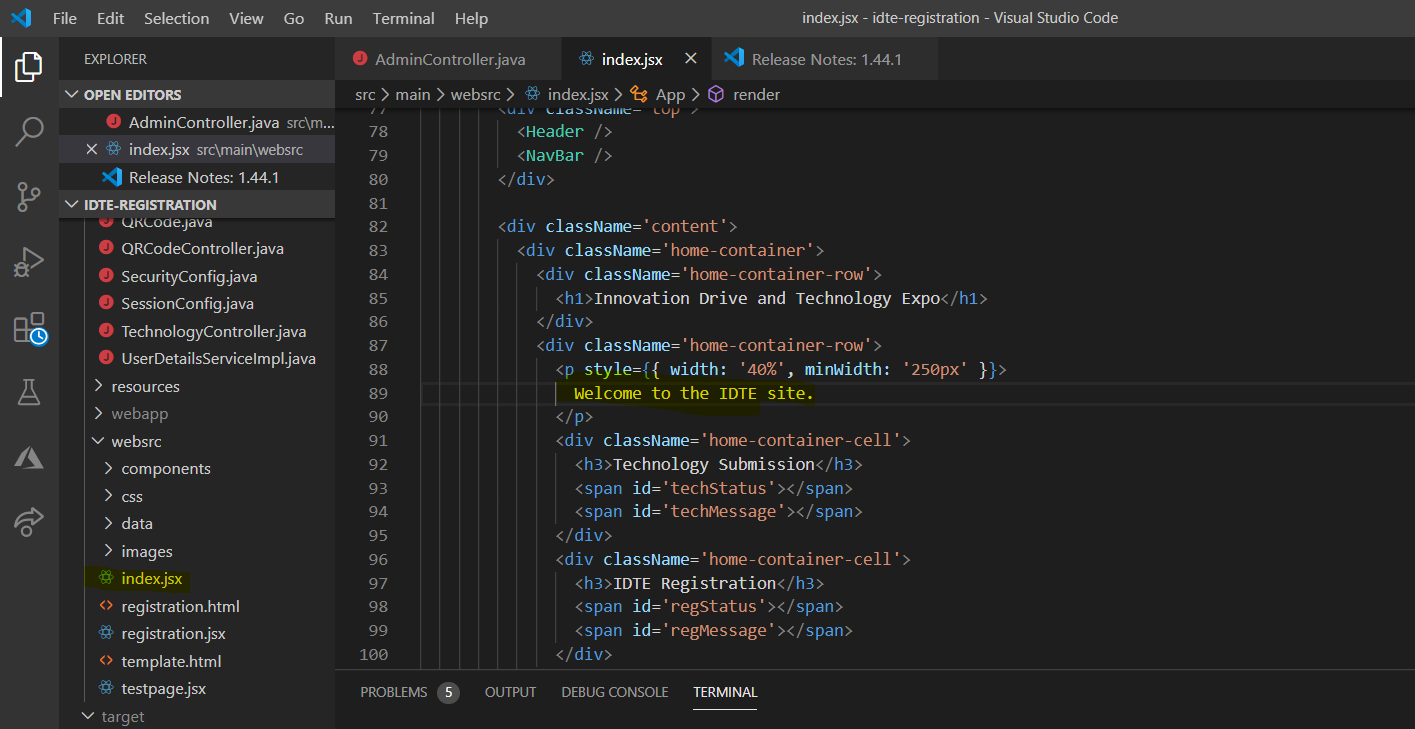
For a user who would need to run it locally on their computer. The following programs are needed:

* Visual Studio Code
* Postman
* Microsoft SQL Server Management Studio
* Microsoft SQL Server

## Customizing product

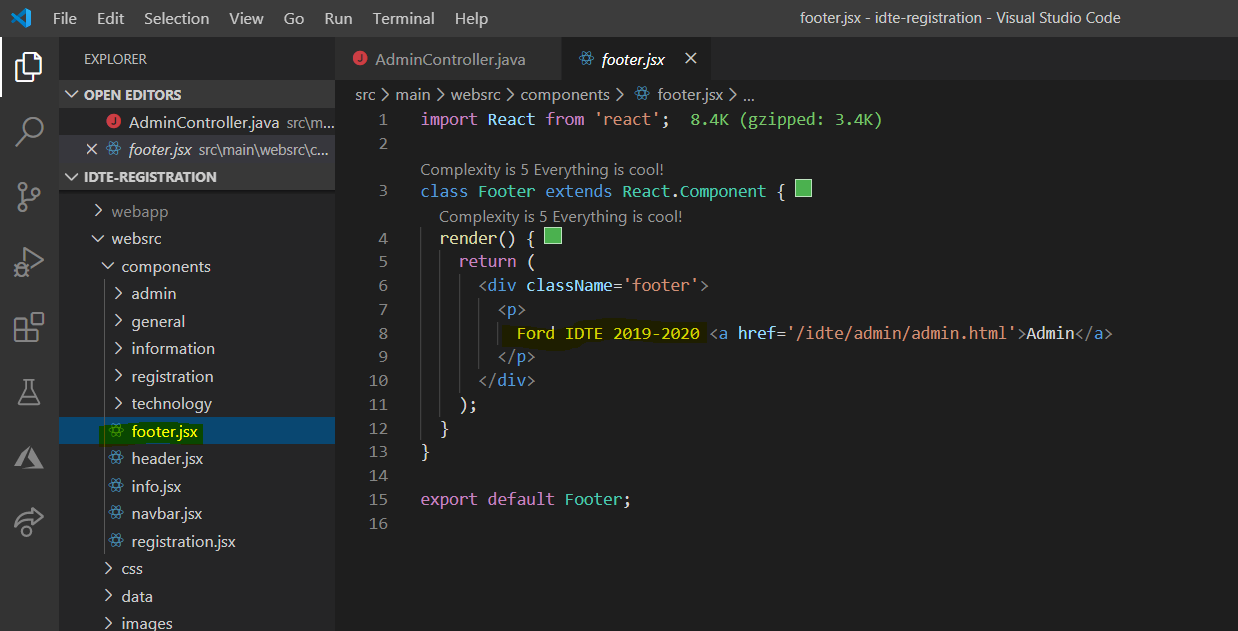
This program can be customized using Visual Studio Code since we have not allowed our client a way to edit the site on a front-end user perspective. Below are the **files** that our client can **edit** and where or what to do with them in order to prevent any fatal errors to happen on the site. There are also screenshots to show specifically what can be changed. If any changes are made below, the command **npm run build** would need to be run to update the frontend changes.

### Home Page



In the above screen shot, the **file name** that would need to be accessed is index.jsx. It can be found through the following pathway: src \ main \ websrc \ index.jsx. The **location** that can be edited is what is highlighted on the right hand side. Inputting the information between the two “p” signs would update the text next to the Technology and Attendee fields.

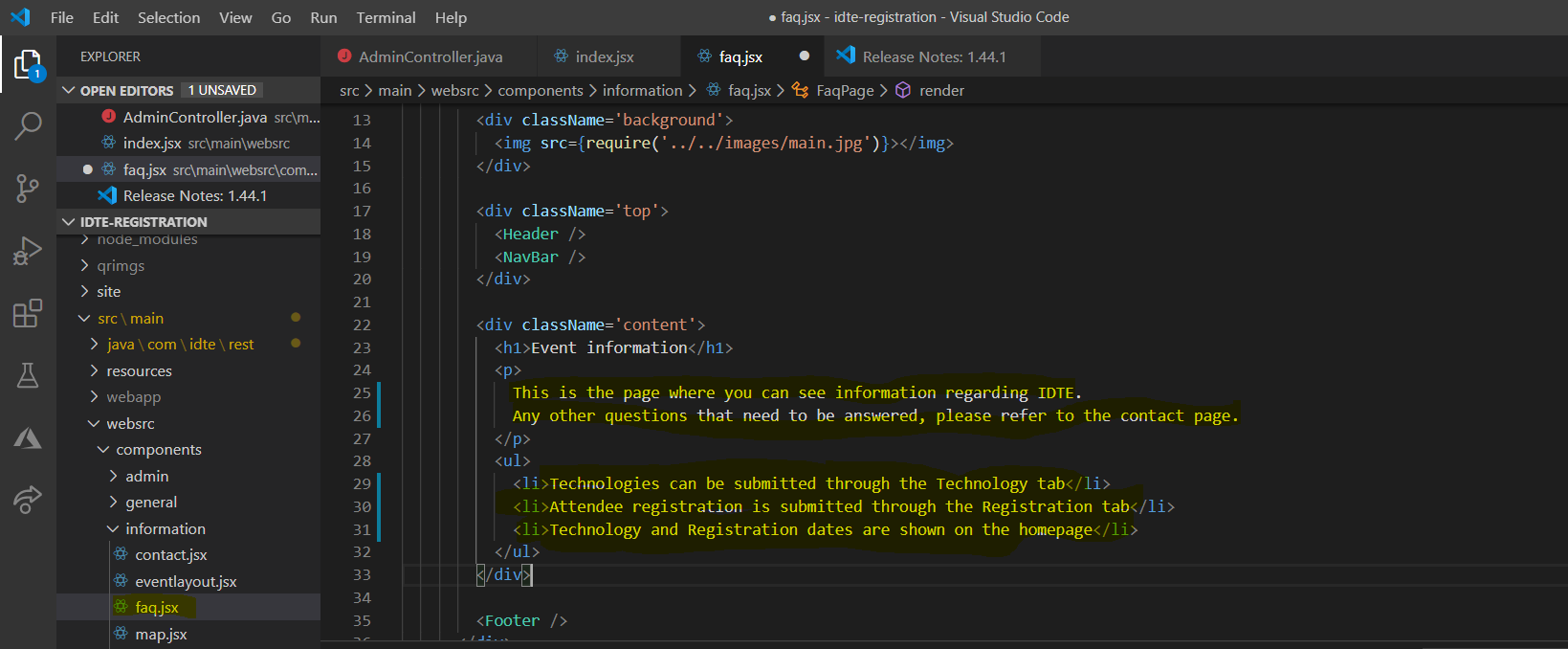
### Footer



In the above screen shot, the **file name** that would need to be accessed is footer.jsx. It can be found through the following pathway: src \ main \ websrc \ footer.jsx. The **location** that can be edited is what is highlighted on the right hand side. Inputting the information between the two “p” signs would update the copyright date. **This is the only portion that needs to be changed. Selecting the “<a href>” field is not needed.** That section is used to have a link to the admin login page.

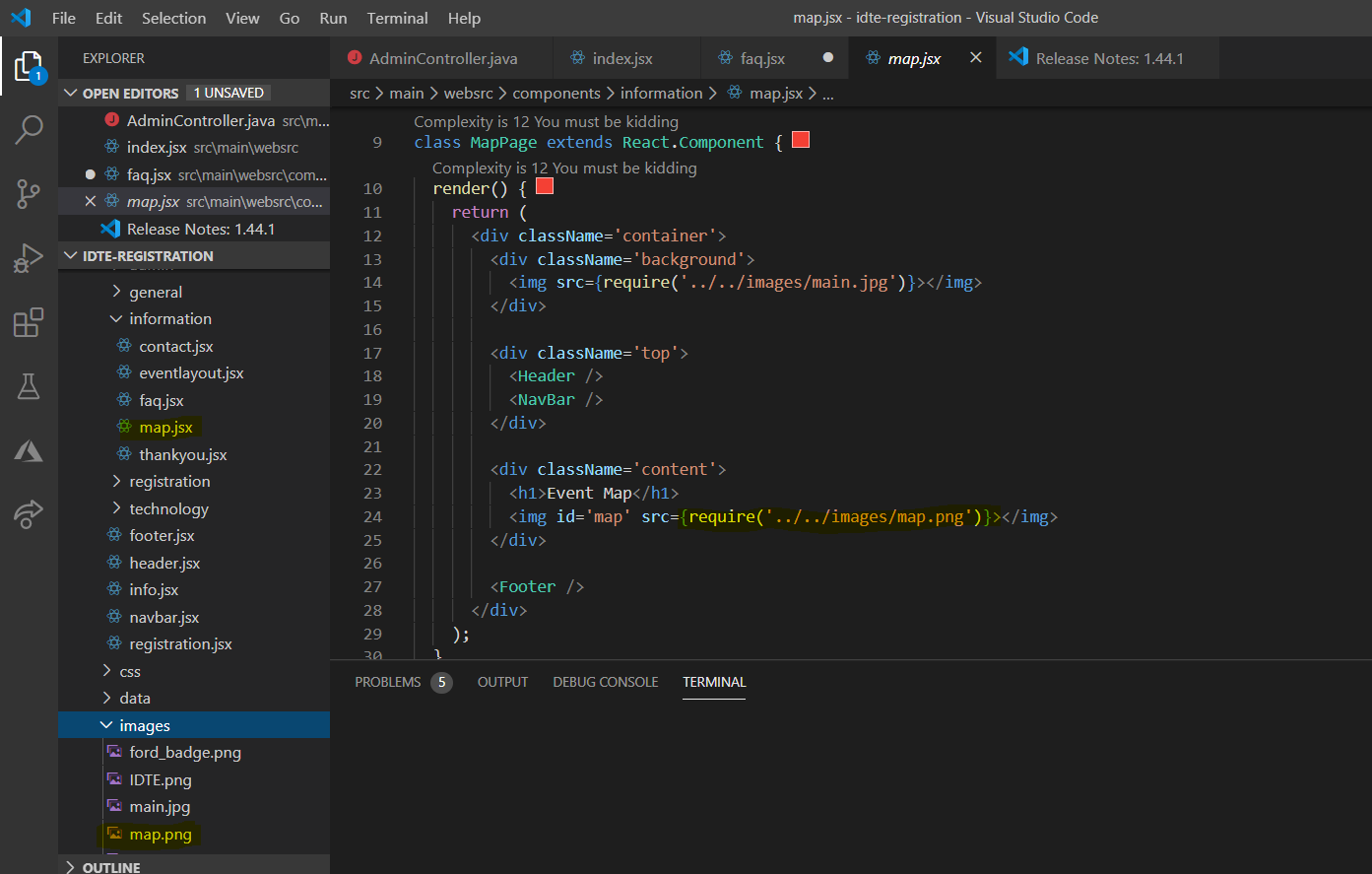
### Information Page

#### Event Info / FAQ



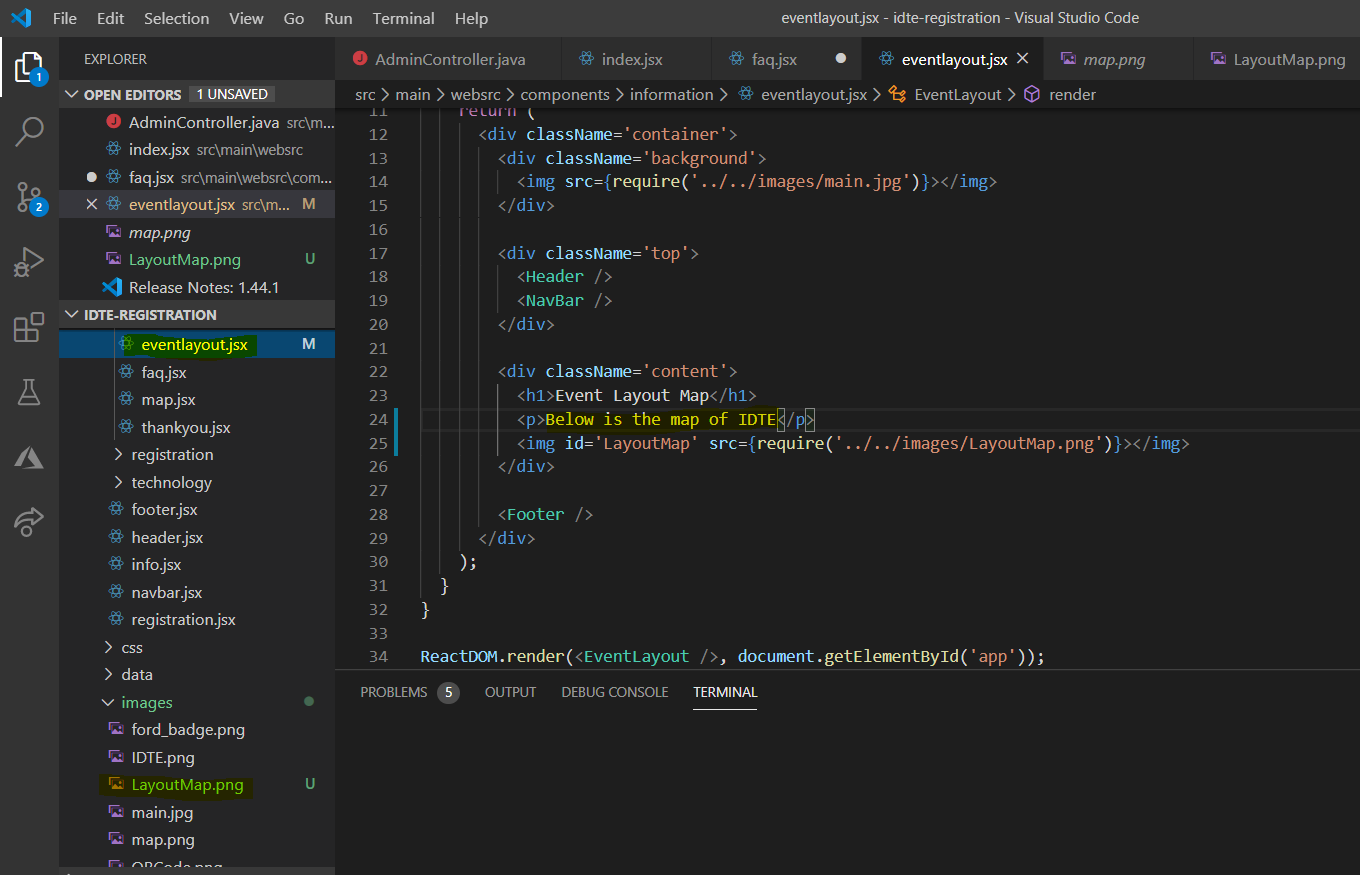
In the above screen shot, the **file name** that would need to be accessed is faq.jsx. It can be found through the following pathway: src \ main \ websrc \ components \ information \ faq.jsx. The **location** that can be edited is what is highlighted on the right hand side. The “p” sections would be used to fill out items in a paragraph format. If there are more bullet point sections to be added underneath, create a new open and close of “li” below the last “li” row.

#### Event Map



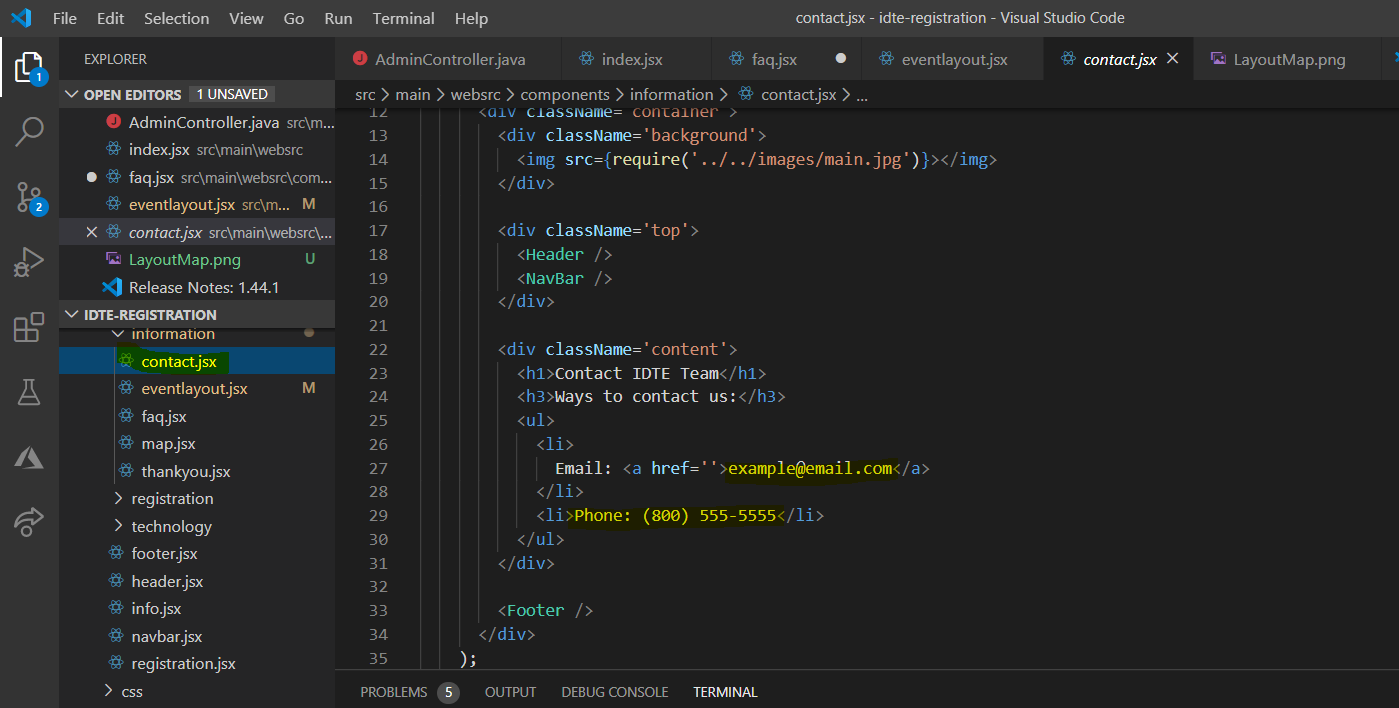
In the above screen shot, the **file name** that would need to be accessed is map.jsx. It can be found through the following pathway: src \ main \ websrc \ components \ information \ map.jsx. This file is different with editing since it **does not require** changes to the file itself. If a new image wants to be added, place it in the file pathway of src \ main \ websrc \ images \ map.png. Overwriting the png map image in this folder would update the image that is shown on the Event Map page.

#### Event Layout



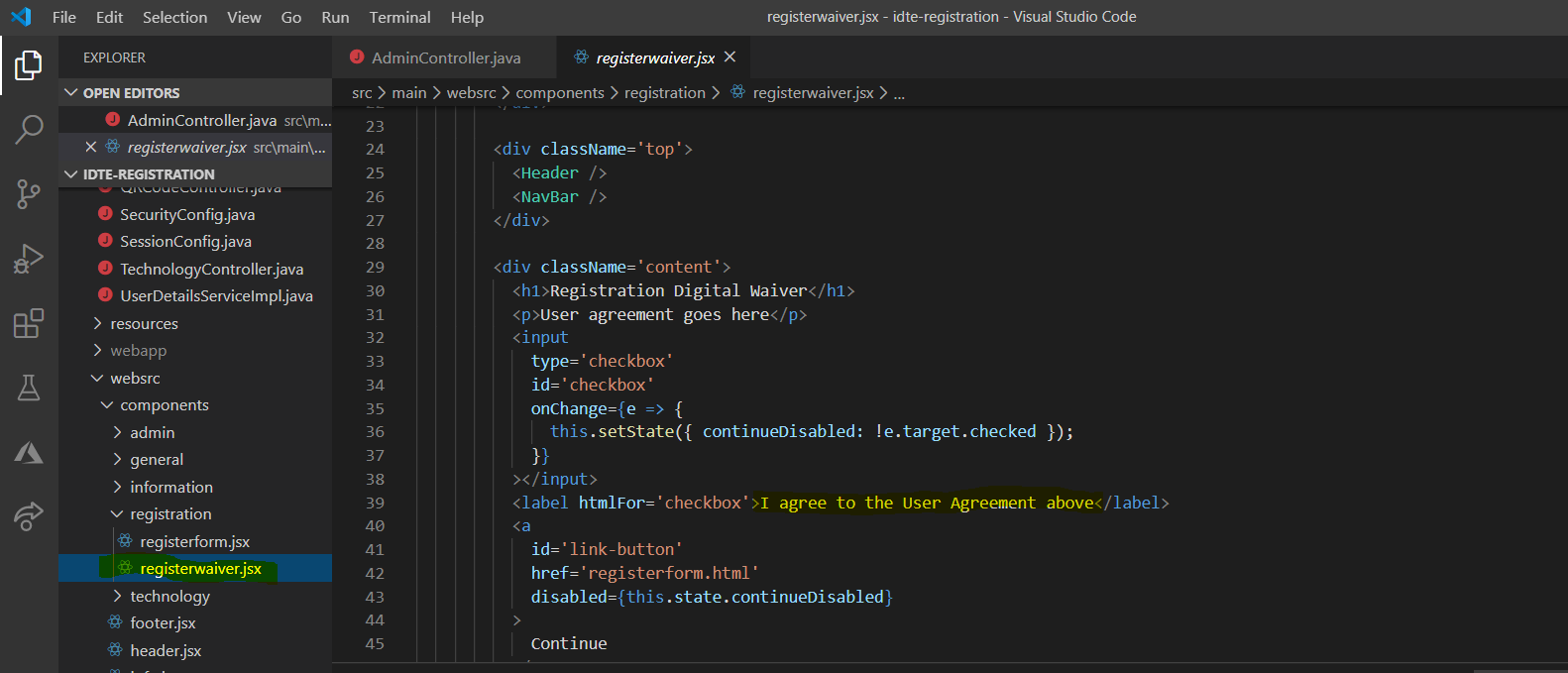
In the above screen shot, the **file name** that would need to be accessed is eventlayout.jsx. It can be found through the following pathway: src \ main \ websrc \ components \ information \ eventlayout.jsx. This file is different with editing since it **does not require** changes to the file itself. If a new image wants to be added, place it in the file pathway of src \ main \ websrc \ images \ LayoutMap.png. Overwriting the png LayoutMap image in this folder would update the image that is shown on the Event Layout page.

#### Contact IDTE



In the above screen shot, the **file name** that would need to be accessed is contact.jsx. It can be found through the following pathway: src \ main \ websrc \ components \ information \ contact.jsx. The **location** that can be edited is what is highlighted on the right hand side. We have “li” sections created since this would be displayed in a bullet point format. If you choose to add more ways of contact, add more “li” sections underneath the latest “li” section.

### Digital Waiver Page



In the above screen shot, the **file name** that would need to be accessed is registrationwaiver.jsx. It can be found through the following pathway: src \ main \ websrc \ components \ registration \ registrationwaiver.jsx. The **location** that can be edited is what is highlighted on the right hand side. Inputting the information between the two label fields would update what is shown on the Attendee Registration Waiver page. **This is the only portion of the file that needs to be edited**.

# Tutorial

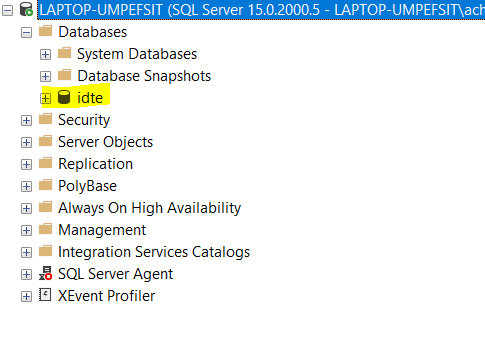
## Setup the program locally

### Visual Studio Code

On Visual Studio Code, to get the program started up **for the first time**, open a new terminal under the Terminal → new terminal. On the terminal, enter the command **npm i**. This command would install all npm file types to the project. This is needed to run the site locally. Once this install is run, run the command **npm run build** to generate the latest website. Finally, select “Application.java” underneath src \ main \ java \ com \ idte \ rest. Right click and select “Run”. This will run our site locally. Once completed, open any web browser and input the command “localhost:8080/idte”. This will open the site locally and any functionality can be used.

### Microsoft SQL Server Manager

* + - 1. Steps for Setup:
* Create a database called **idte**



* Under Logins, create a new login with the following credentials:
  + **Username:** idteroot
  + **Password:** idteroot123
* Create a new query in SQL Server Management
* **Enter the below script in its entirety as shown**

CREATE TABLE SPRING\_SESSION (

PRIMARY\_ID CHAR(36) NOT NULL,

SESSION\_ID CHAR(36) NOT NULL,

CREATION\_TIME BIGINT NOT NULL,

LAST\_ACCESS\_TIME BIGINT NOT NULL,

MAX\_INACTIVE\_INTERVAL INT NOT NULL,

EXPIRY\_TIME BIGINT NOT NULL,

PRINCIPAL\_NAME VARCHAR(100),

CONSTRAINT SPRING\_SESSION\_PK PRIMARY KEY (PRIMARY\_ID)

);

CREATE UNIQUE INDEX SPRING\_SESSION\_IX1 ON SPRING\_SESSION (SESSION\_ID);

CREATE INDEX SPRING\_SESSION\_IX2 ON SPRING\_SESSION (EXPIRY\_TIME);

CREATE INDEX SPRING\_SESSION\_IX3 ON SPRING\_SESSION (PRINCIPAL\_NAME);

CREATE TABLE SPRING\_SESSION\_ATTRIBUTES (

SESSION\_PRIMARY\_ID CHAR(36) NOT NULL,

ATTRIBUTE\_NAME VARCHAR(200) NOT NULL,

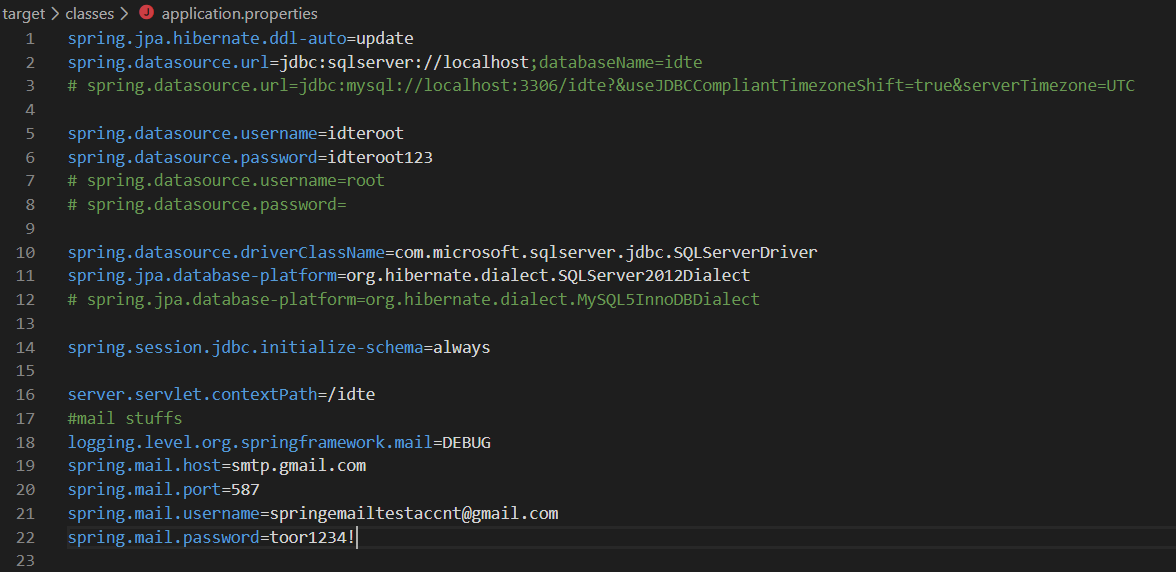
ATTRIBUTE\_BYTES IMAGE NOT NULL,

CONSTRAINT SPRING\_SESSION\_ATTRIBUTES\_PK PRIMARY KEY (SESSION\_PRIMARY\_ID, ATTRIBUTE\_NAME),

CONSTRAINT SPRING\_SESSION\_ATTRIBUTES\_FK FOREIGN KEY (SESSION\_PRIMARY\_ID) REFERENCES SPRING\_SESSION(PRIMARY\_ID) ON DELETE CASCADE

);

* Once the script is entered, select the Execute button on SQL Server Management. The tables will now be generated.
* In the application.properties file (found on VS code) make sure the following is setup as shown in the below screenshot. To access this file, the pathway is target \ classes \ application.properties



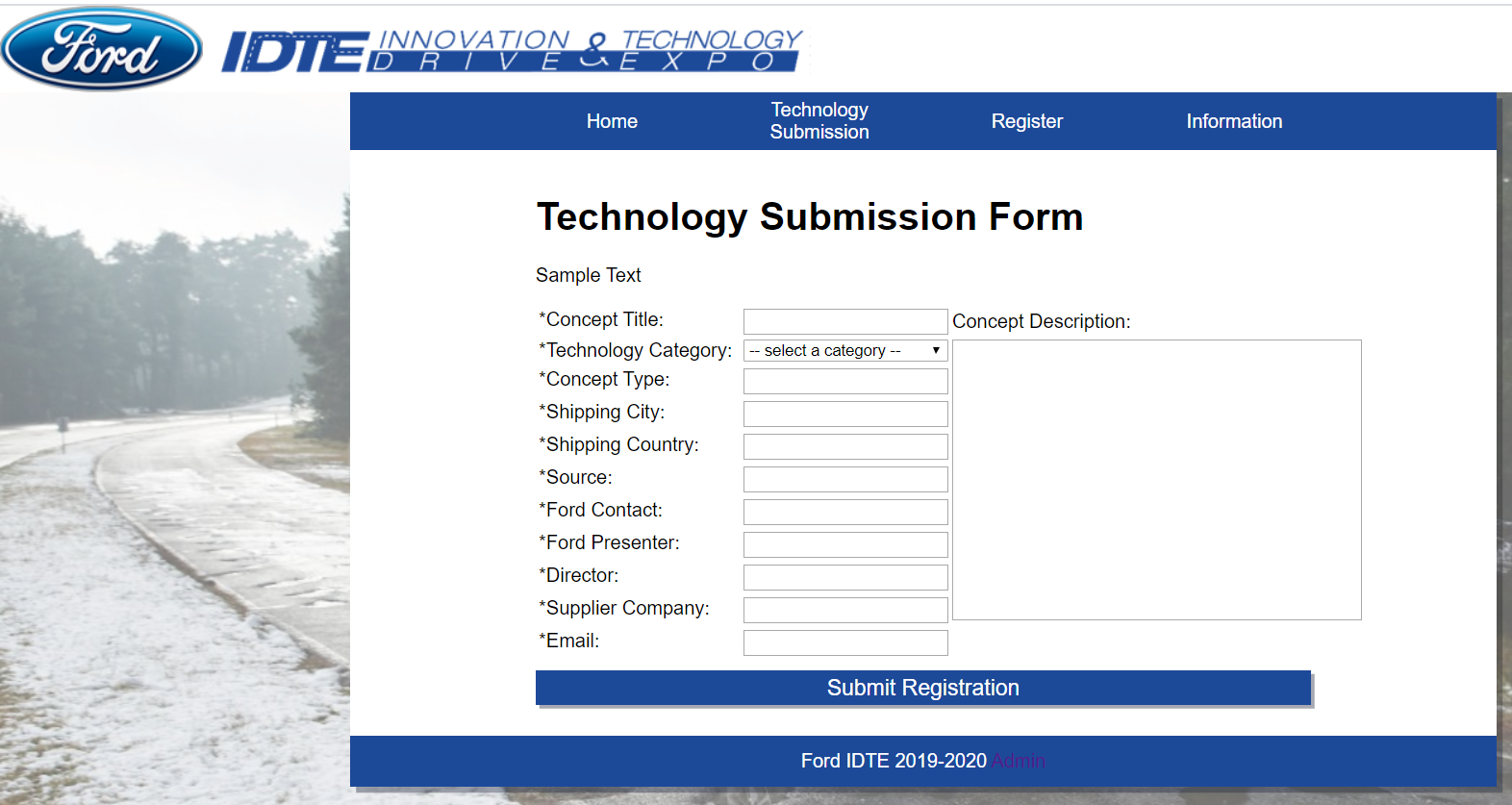
## Walkthrough of example

### Be an attendee and go through the steps

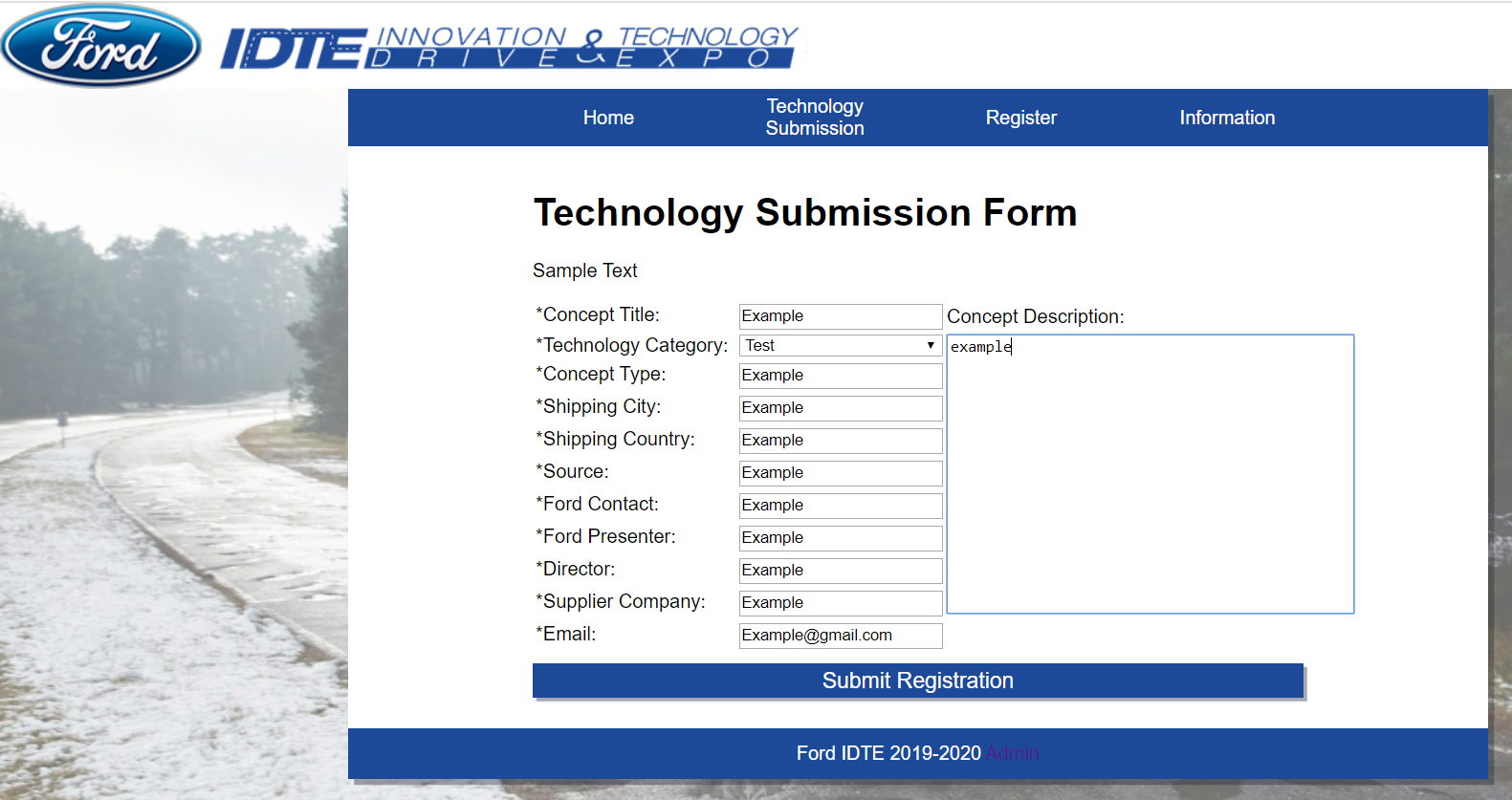
The below steps and screenshots are used to show the steps taken for user registration. Please note, the technology submission and registration buttons cannot be accessed if registration for either one is currently disabled. This is controlled through the Event page that only an admin has access to.



In the attached screenshot, I put in the URL for IDTE and I am shown the homepage.



Next, I select the “Technology Submission” button on the navigation bar and I am taken to the Technology Submission Form.



I filled out the Technology Submission Form with credentials that were requested of me. Please note, all fields are required for this submission.



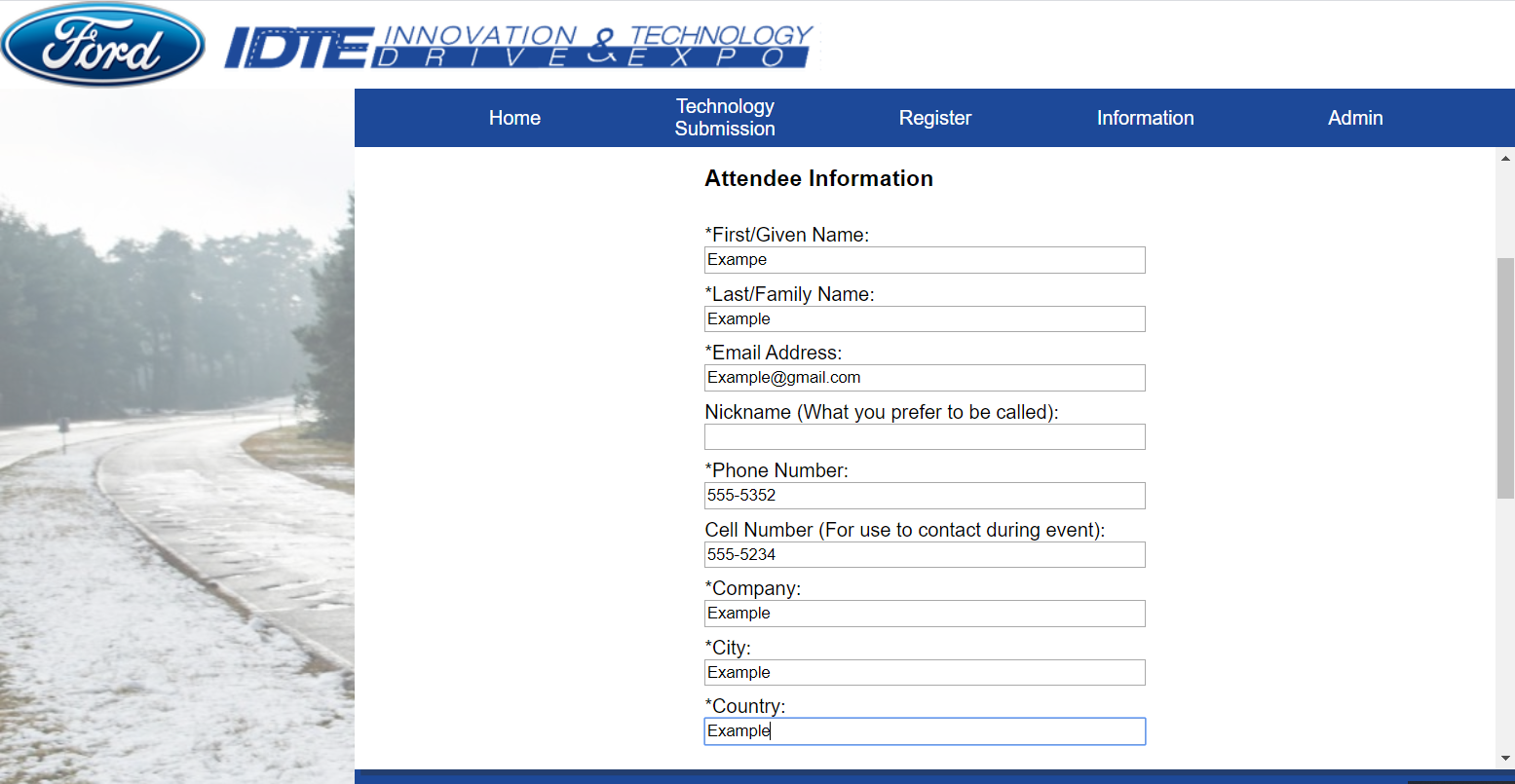
I then select the Submit Registration button and am now shown this confirmation page. Behind the scenes, I would be getting an email sent to the account that was placed in the “Email” field



I select the Registration button and am shown the Registration page.

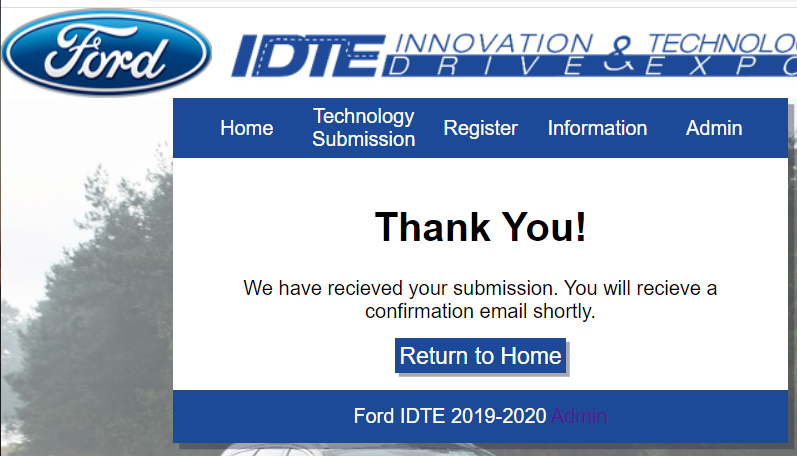


I select Supplier and am now shown the Digital Waiver page





Agreeing to the waiver takes me to the registration page. Here I fill out my information and select days I want to attend for IDTE. Please note, when selecting Evaluator, we can only select one of five event days and select our top 3 Technology Categories.



I select the Submit Registration button and am shown a confirmation page. An email would be sent to me with a QR code of my attendee ID.

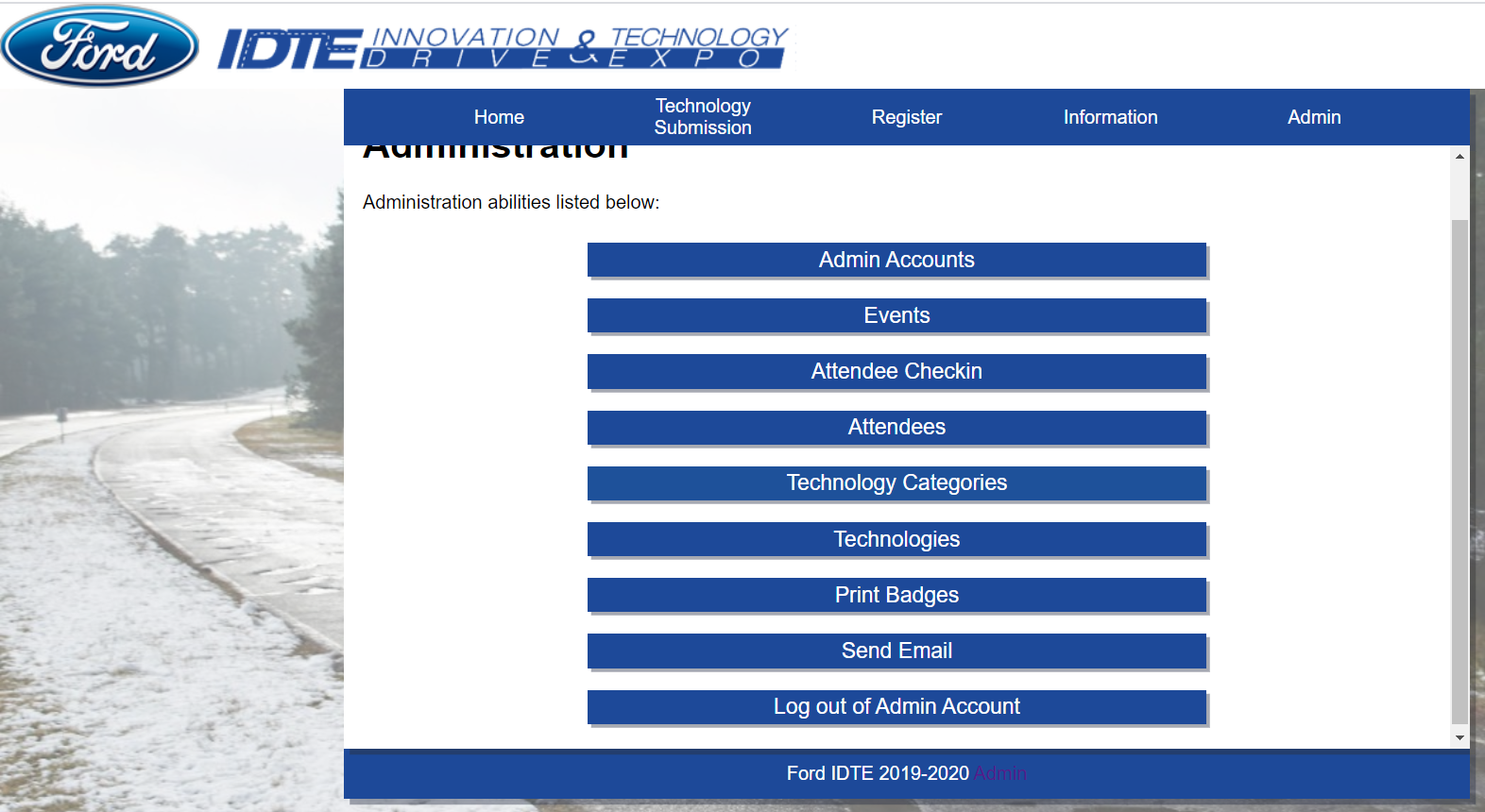
### Check In the Attendee as an Admin



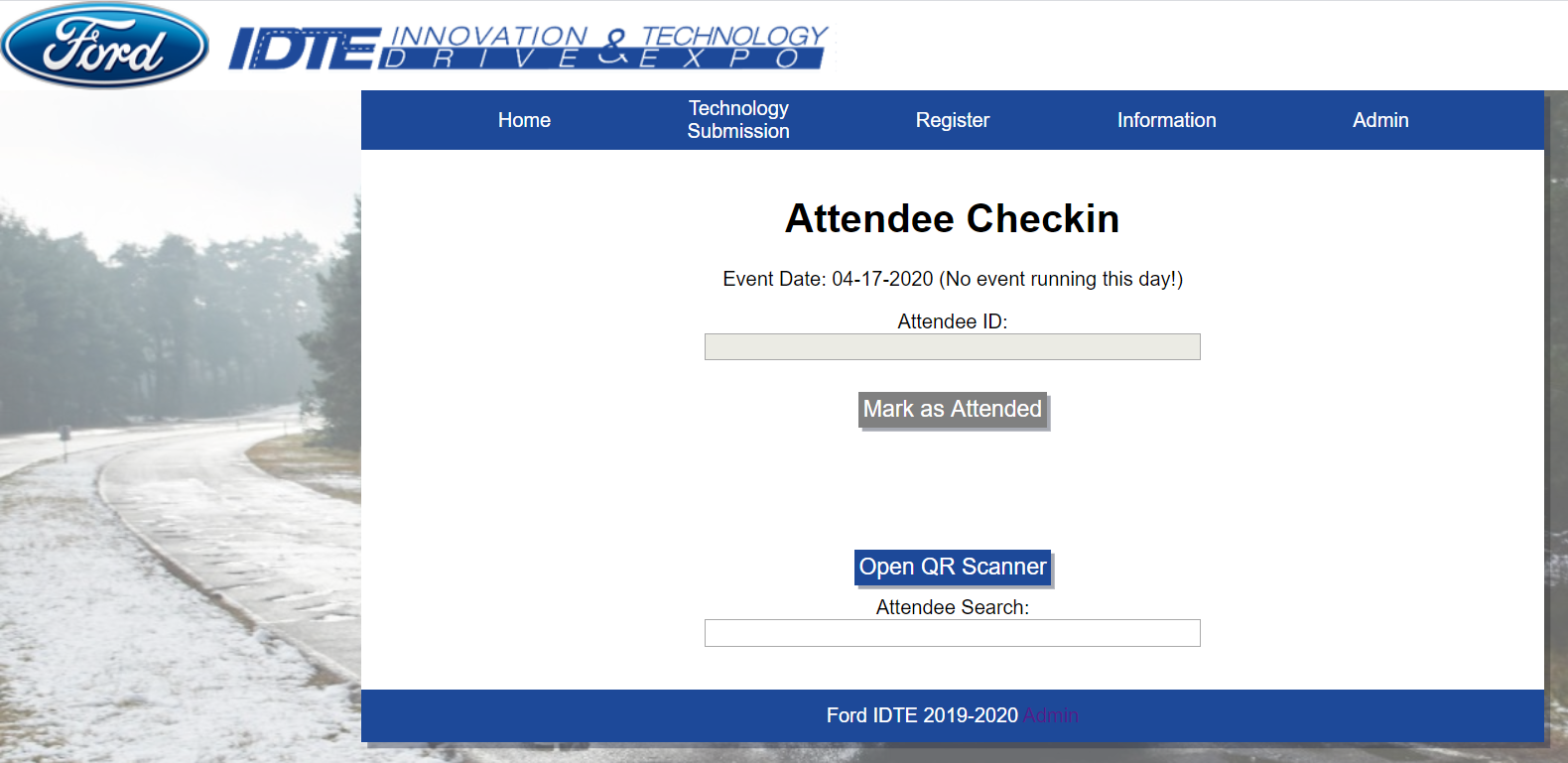
I open the webpage URL and am shown the homepage for the site.



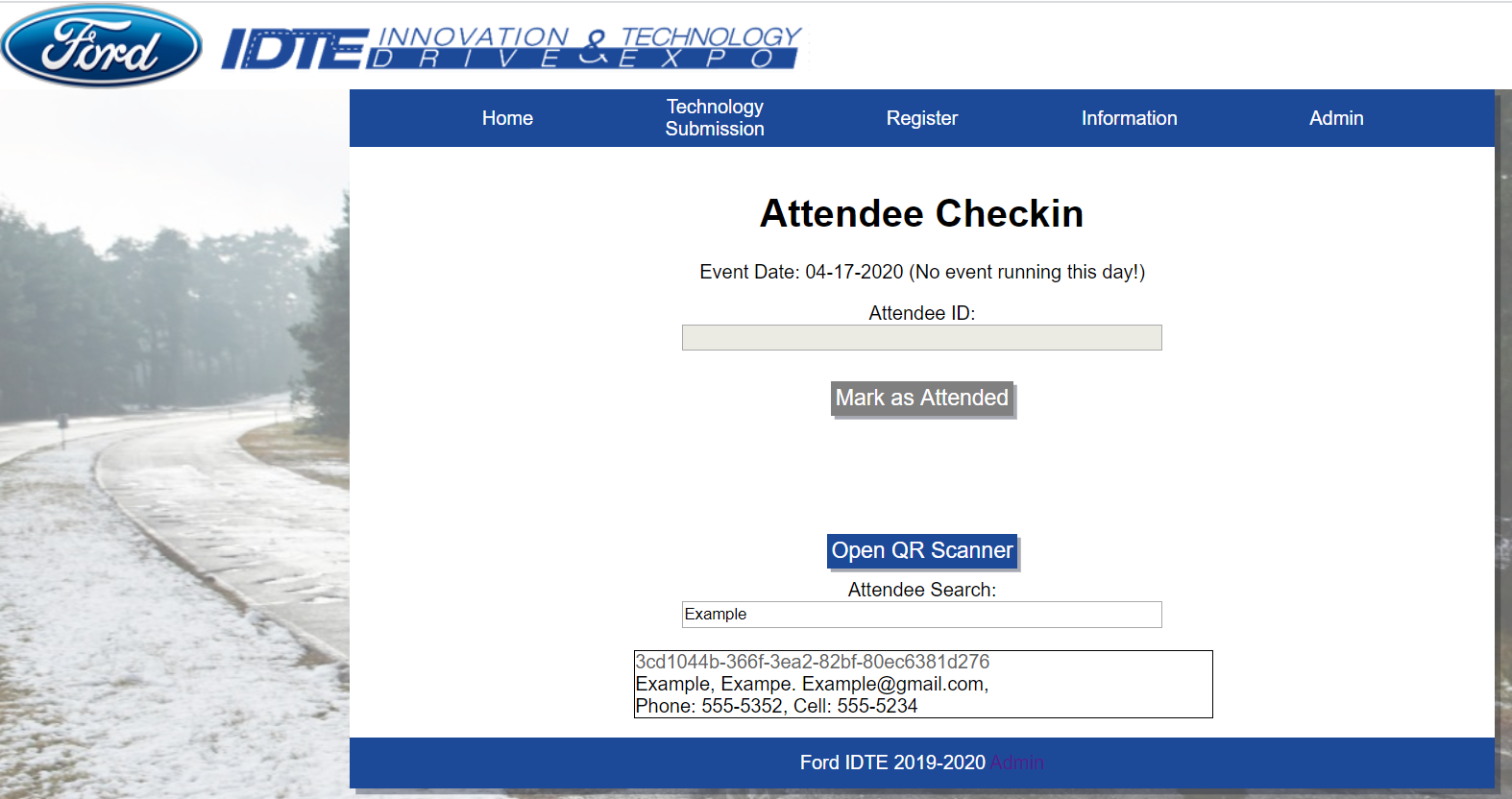
Selecting the admin hyperlink button next to the copyright, I am taken to a user login page. My credentials are placed and remembered. In the above, I am using the admin login. So the **username** is admin and **password** is password.



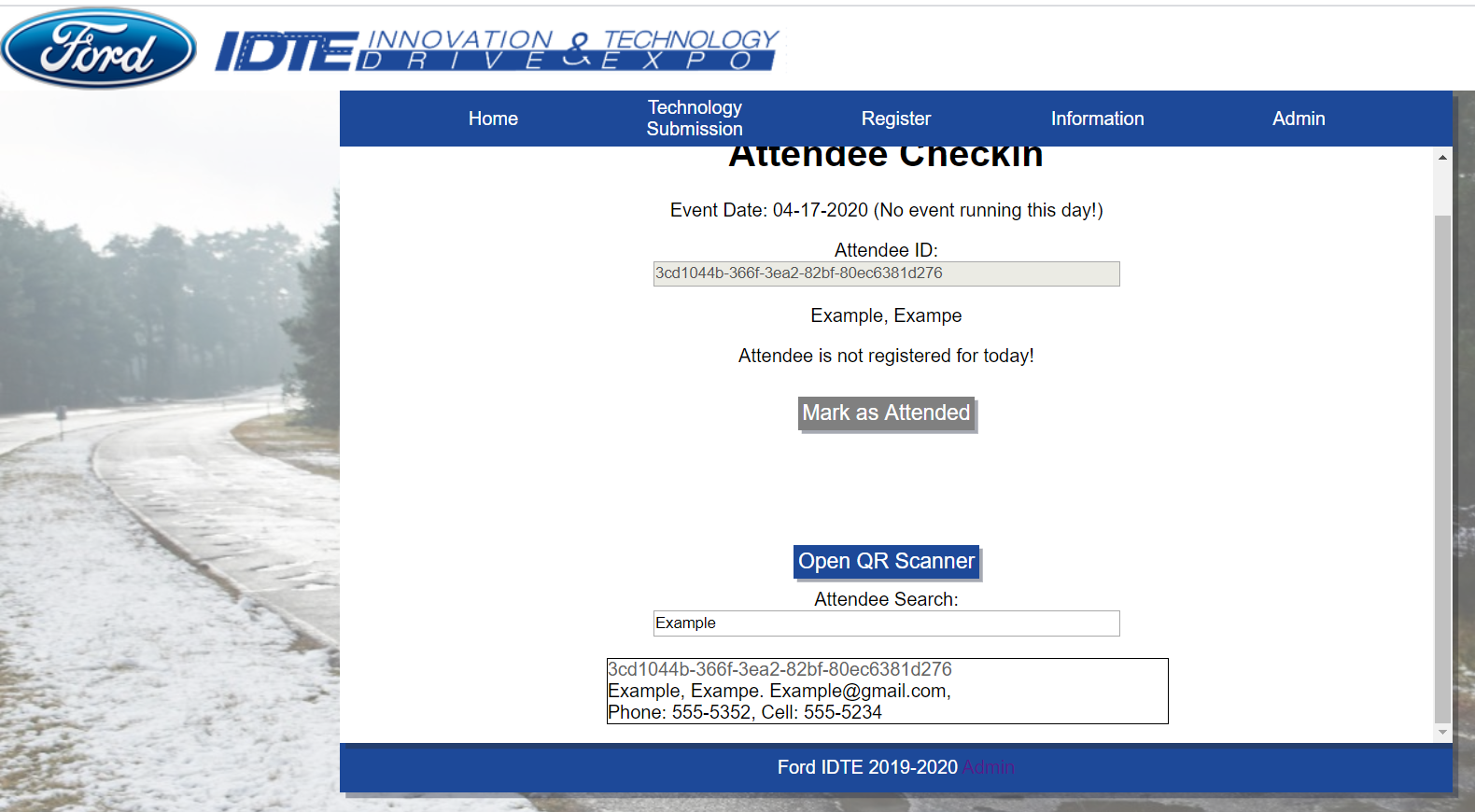
I select the Sign in button and after a valid sign in is received, I am taken to the Administration view. Please note the admin panel being created on the header now since I am logged in as an admin.



I select the Attendee Checkin button and then the Scan QR code button. I am taken to the following page. Here I can checkin attendees on a given day for the event.



I input the attendee name I created in the last example into the text field and exit out of it. As you can see, below is a list of all attendees that match to what is entered in the text field.



Selecting the attendee in the list generated the Attendee ID in the Attendee ID field. Since we are currently on a day that our example Attendee should not be at, they cannot be marked as attended. Once the attendee is checked in on a day that they are expected to be there, they can be marked as attended.

## Explanation of example

The above examples are used to show the flow of events that would take place when you are a normal user going through registration. The other example is used to go through admin rights and permissions. Both of these are used since we have two different kinds of users that would be using this site.

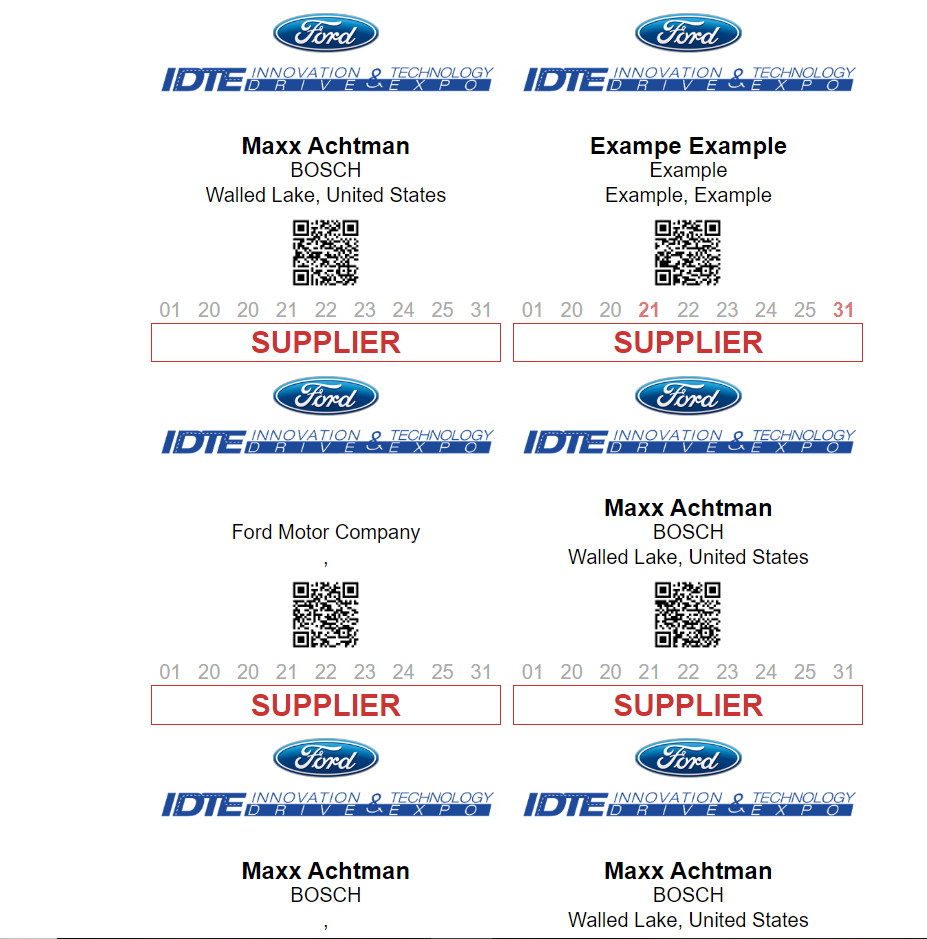
## Using help (on-line & text)

Please refer to section 2 for the help portion. This will contain all things that are needed to run the program as expected. If there are any errors that may occur please refer to the attached LOG file that Visual Studio Code would create. In there, you would find an error number that can be used on Google. The error would have something related to the device, i.e. a missing jre or jdk file type.

# Detailed Instruction

## Output from product

Here is the final output of our product. The end goal is to have badges printed out that our attendees can use during the days of the event.



These are printer friendly badges. Each page has 6 badges created. On each badge, we would have the name, company, city, country, QR Code, and dates they would be there in attendance. Those marked in red are dates that the attendee is expected to be there. The dates listed would include the three set up days, the dry run day, and the five event days.

## Input to product

Please see section 3.2 of the input for the product. There, we cover registration for technologies and attendees. These are the expected inputs that we would receive for the website.

## Operation of product

During the process of our project, a few different features would happen. One example of this is our website would send an email to the person who registered a technology or to attend the event. Also, during attendee registration a QR code would be created and will hold the attendee id that is automatically generated during the process of adding the attendee to the database.

## Error handling

Our program does not have any fatal errors occuring. If errors may occur, they can be seen during initial running of the program and can reference section 2 for support handling where that may come up.

## Specific functions

### How to run

Please see section 3.1 to learn how to run the program. Since we cannot launch the site through Ford’s servers we cannot provide another way to run our program outside of the steps provided to run it locally.

### Input data needed

Input data that is needed to run our site can be found in section 3.2 where we provide steps for registration. These are the fields that require input from the user. We also have the superadmin (primary admin) account already created for our site with login credentials.

### Output interpretation

Please see section 3 for output interpretation. We are generating emails to the users that can be seen once entering their own email credentials. The other outputs we offer are printer friendly badges and confirmation pages.

# Technical Details

## Principles of operation

The main operation of our site is to run it through Ford servers. Since there are specifics needed for security reasons, we have to follow their guidelines. That being said, this has yet to be launched to Ford servers for understandable reasons. Passing this to our client for their use, we are ensuring that on a **local** perspective, every major function works as expected. This is known to the client.

## Advanced features

Our website can be used on both mobile and PC with any browser. The website is formatted to fit a mobile screen and any functionality seen on PC can be done on mobile. There are also alternatives taken as well, including giving users the option to scan QR codes through an image. **Please note** this feature would only happen if there is not a camera available on the device or it cannot load.

## Modified of product

This is an updated version of a project from an earlier team. What we did was rebuild it from the ground up with the software requirements specified by Ford. We have allowed more user friendly functionality at the click of a button. The admin users on this site have full reign over controlling Event dates, technologies, and attendees.

## Support information

Any support information can be found within this document. This document is covering the basic functionalities of our site and how to edit it on their own local PCs since this is a site that will be used yearly and would therefore require updates.