# **INTRODUCTION**

Large numbers of incidents are raised in almost all the corporate companies these days. Resolving all such incidents and making a note of all such incidents is a tedious and time consuming task for both the administrative and the staff of such companies. The aim of this project is to provide a Service Incident Productivity Tracker, a web application that is able to manage the creation, updating of incidents and movement of tasks from desk to desk of personnel who work on them. The incident has a priority and a status based on which the person to whom the task is assigned to can perform the tasks in an orderly manner and change the status to distinguish the tasks that are accomplished from those which are not. The system was developed with open source software after establishing vital functional and non-functional requirements. Files such as, reports, decisions, requests, reminders can be processed and tracked by the system in real time. The system is interactive and usable and able to improve Incident Management and productivity in corporate company production operations.

## **Incident**

An incident is an unplanned interruption or quality reduction of an IT service. The service level agreements (SLA) define the agreed-upon service level between the provider and the customer.    
  
Incidents differ from both problems and requests. An incident interrupts normal service; a problem is a condition identified through a series of multiple incidents with the same symptoms. Problem management resolves the root cause of the problem; incident management restores IT services to normal working levels. Requests for fulfillment are formal requests to provide something. These may include training, account credentials, new hardware, license allocation, and anything else that the IT service desk offers. A request may need approvals before IT fulfills it.

Incidents interrupt normal service, such as when a user’s computer breaks, when the VPN won’t connect, or when the printer jams. These are unplanned events that require help from the service provider to restore normal function.

Some of the basics of Service Incident management are,

## **Incident vs service request**

Service requests are a formal request from a user for something to be provided – for example, a request for information or advice. The main difference between Incident and service request is that often pre-approved standard changes are classified as service requests which end users request for. For example, UX designer requests for Adobe Photoshop software and increase in RAM space. Having an intuitive service catalog to capture this request is recommended.

## **Incident v/s problem**

Problem is a series of incidents with an unknown root cause, whereas incident arises as soon as something breaks or stops working disrupting normal service. Incident handling is usually a reactive process whereas problem management is more proactive. Incident management system aims at restoring services quickly whereas problem management aims at finding a permanent fix.

# **IMPLEMENTATION**

## **Requirements**

## **2.1.1 Software Requirements**

Visual Studio 2017, Microsoft SQL Server Management Studio

## **Project Implementation Details**

### 2.2.1 Database Creation

The world of data is constantly changing and evolving every second. This in turn has created a completely new dimension of growth and challenges for companies around the globe. By accurately recording data, updating and tracking them on an efficient and regular basis, companies can address their challenges on one hand and make use of the immense potential offered by this sector on the other hand.

Similarly, a database is created for the purpose of storing the incidents and their characteristics. An incident will have many features like the user to whom the incident is assigned to, and the area of the incident along with its priority of closure. In order to maintain a clear understanding and relationship between all the characteristics of the incidents, we have created a database with some tables.

The tables and the features of the tables in the database used for this project are,

**User:**

This table contains the data of the users and the administrators. The user table contains the user names and the corresponding password of the user. The person is given a Role Id based on the responsibility. The Administrators are given the Role Id of 1 and the users are given the role Id of 2. These role ids are the foreign keys from the Role table. The User table also contains the details of the created time of the incident and the time it got modified. These are used as it is a better practice to analyze the capability of the employee working on the incident.

**Role:**

This table contains the columns of Roles and the respective role Ids, the administrators and users have role ids 1 and 2 stored in the table. They are used in other tables using the foreign key constraints. The Role table has a description column which has the details of the operations to be done by the employee. This table also has the columns of Created Date Time and Modified Date Time as it is a good practice to know the date and time of any changes made in the table.

**Area:**

The Area table in the database represents the area to which the particular incident belongs to, for example, Sales, Marketing, Installation error, etc. It contains the Area\_Id, which is different for every area and it is an identity number which automatically gets incremented. This is used to access the different attributes of the area table from other tables in the database. As it is an identity number it is a primary number in default, so it is preferred to use it than any other attribute. The table has Area Name which is the department name along with the description column which gives a brief description about the category of the incident. This table also has the created and modified date and time to know when some change is made to the table.

**Priority:**

This table contains the data of the priority of the incident. Some incidents can have high importance and need to be resolved as soon as possible, such incidents are given the highest priority and the incident which can be solved later on are given some low priority, all such details are stored in this table. The priority table has Priority Id which is a primary key with the corresponding priority name like High, Medium, Low. The priority can be changed once the incident is initiated so for making an account of the priority changes of the incident, the columns of created and modified date and time are maintained.

**Status:**

The status table in the database gives the details and information about the progress of the incident logged. Each incident goes through many phases like being logged, being assigned to the user and getting resolved. Once the incident is solved it is closed. All such details are maintained in this table. This table has Status\_Id which is used to access the values in other tables. The status name and description columns gives the interpretation of the incident so that both the end user and the employee can understand the importance of the incident. The date and time created and modified are maintained to analyze when the status has been changed.

**Incident:**

Incident table is the most important table in the entire database as almost all the operations are performed on this table. This table has attributes specific to it and some attributes that are taken from other tables in the database. This table contains the Incident Number, which is an identity number which gets incremented whenever a new incident is added. Each incident is given an Incident Name so that it is easy to interpret the problem raised, along with the description so that the end user can explain the problem to the employee who is going to solve the issue. Each incident belongs to an area and it is specified in the table so that it is easy for the administrator to assign the task to the user who is capable of resolving the issue. The table also has Area\_Id which is taken from the Area table and the Priority\_Id from the priority table and User\_Id from the User table. All these values help in coordinating the details. Target\_Date is the column which specifies the employee to finish the incident raised by that date, it can be changed. The status\_id gives the importance of the incident and it is taken from the Status table. The Created and modified date and time are placed in the table to maintain an account of when and how the incidents are resolved.

## **Incident Management Process Flow**

### Incident Logging

The first step in Incident management is to report the identified incident. This can be done by the end users themselves or agents can do it on their behalf. The IT team needs to capture complete information about the incident using a form template to speed up recovery process. They also need to set up relevant channels for end users to report an issue easily.

**2.3.2 Incident Classification**

Segment the incidents with appropriate category/sub-category in order to easily identify the right group and agent. Customize incident form with the right fields and set up automated rules for ticket classification, prioritization and assignment and save valuable time in the process. Correct classification of Incidents will help in generating reports faster.

**2.3.3 Incident Prioritization**

Assigning the right priority to ticket has a direct impact on deciding SLA policy and addressing business critical issues on time. Thus, set up a realistic SLA definition to meet customer commitments.

**2.3.4 Investigation and Diagnosis**

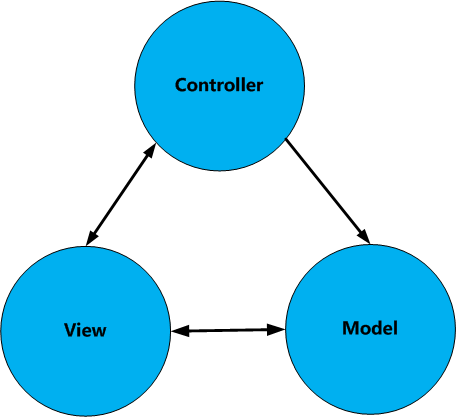
When an incident is raised, the IT team performs an initial analysis of the same and sends a resolution to the end user. In the event the resolution is not available immediately, they, escalate the incident to tier II & tier III teams for detailed investigation. Components required to identify, analyze, and contain an incident are reviewed. The incident is also associated with the relevant CI (Configuration Item) for faster diagnosis.

**2.3.5 Incident Resolution and Closure**

One of the primary goals of any IT team is to resolve any incident, coming their way, as soon as possible. Efficient communication about the resolution and closure of the resolved tickets is very important. The team can even automate the process of closing the resolved tickets.

**2.4 Model-View-Controller**

ASP.NET MVC is an open source platform for building web applications using the MVC architecture introduced by Microsoft. MVC is nothing but Model-View-Controller and initially was called by the name “Thing-Model-View-Editor".



**Figure 2.1:** Model-View-Controller

The MVC design pattern is defined in three layers like Model, View and Controller.

* **Model**: It is a collection of classes or library files for the business layer with a specific domain functionality. The business layer contains a set of business rules, entities and components for data manipulation purposes.
* **View:**It is a representation of an application user interface (in other words Presentation layer).
* **Controller:**It is a collection of classes that handle interaction from the user request and the response to display the information to the view.

**2.4.1 MVC Design Pattern**

**Top of Form**

**Bottom of Form**

**ASP.NET MVC** is an open-source software from Microsoft. Its web development framework combines the features of MVC (Model-View-Controller) architecture, the most up-to-date ideas and techniques from Agile development and the best parts of the existing ASP.NET platform.

The **Model View Controller** (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer of an application.



**Figure 2.2 :UML Diagram MVC Design Pattern**

* The **Model** contains only the pure application data; it contains no logic describing how to present the data to a user.
* The **View** presents the model’s data to the user. The view knows how to access the model’s data, but it does not know what this data means or what the user can do to manipulate it.
* The **Controller** exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

**2.4.2 Advantages**

* Multiple developers can work simultaneously on the model, controller and views.
* MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
* Models can have multiple views.

**2.4.3 Disadvantages**

* The framework navigation can be complex because it introduces new layers of abstraction and requires users to adapt to the decomposition criteria of MVC.
* Knowledge on multiple technologies becomes the norm. Developers using MVC need to be skilled in multiple technologies.

**2.5 Data Access Layer**

A **data access layer** (**DAL**) in computer software, is a layer of a computer program which provides simplified access to data stored in persistent storage of some kind, such as an entity-relational database.

**2.5.1 Scaffolding**

Scaffolding is a technique used by many MVC frameworks like ASP.NET MVC, Ruby on Rails, Cake PHP and Node.JS etc. to generate code for basic CRUD (create, read, update, and delete) operations against your database effectively. Further code can be edited or customize the auto generated code according to the need.

Using scaffolding, we can quickly generate code that interacts with data models. This feature mainly reduces the amount of time required to build MVC application with standard data operations. This framework uses Data-First Approach for data operations. Using the Scaffold-DBContext command, we can create entities and context class from an existing database.

**2.5.2 Creating the Data Models**

Entity Framework is an object-relational mapping (ORM) framework. It is used to work with relational data as objects, eliminating most of the data-access code which is usually written by the coder. Using Entity Framework, queries can be implanted using LINQ, then retrieve and manipulate data as strongly typed objects. LINQ provides patterns for querying and updating data. Using Entity Framework allows to focus on creating the rest of the application, rather than focusing on the data access fundamentals.

Entity Framework supports a development paradigm called Code First. Code First lets you define your data models using classes. A class is a construct that enables the user to create their own custom types by grouping together variables of other types, methods and events. Mapping of classes to an existing database can be easily done using Entity Framework.

**2.5.3 Entity Framework and References**

By default, Entity Framework is included when a new **ASP.NET Web Application is** created using the **Web Forms** template. Entity Framework can be installed, uninstalled, and updated as a NuGet package.

This NuGet package includes the following **runtime** assemblies within your project:

EntityFramework.dll – All the common runtime code used by Entity Framework EntityFramework.SqlServer.dll – The Microsoft SQL Server provider for Entity Framework

**2.5.4 Entity Classes**

The classes created to define the schema of the data are called entity classes. Each property in the class specifies a column in the table of the database. These classes provide a lightweight, object-relational interface between object-oriented code and the relational table structure of the database.

**2.5.5 Data Annotations**

Sometimes, certain members of the classes have attributes specifying details about the member, such as [ScaffoldColumn(false)]. These are data annotations. The data annotation attributes can describe how to validate user input for that member, to specify formatting for it, and to specify how it is modeled when the database is created.

### 2.5.6 Context Class

In order to start using the classes for data access, you must define a context class. As mentioned previously, the context class manages the entity classes (such as the Incident class and the User class) and provides data access to the database.

The System.Data.Entity namespace is used to access all the core functionality of Entity Framework, which includes the capability to query, insert, update, and delete data by working with strongly typed objects.

The SIPTContext class represents Entity Framework product database context, which handles fetching, storing, and updating Incident class instances in the database. The SIPTContext class derives from the DbContext base class provided by Entity Framework.

A well-defined Data Access Layer is crucial for any business application. Usually a Data Access Layer follows a repository pattern. Repository Pattern is used to create an abstraction layer between data access layer and business logic layer of an application. Repository directly communicates with data access layer [DAL] and gets the data and provides it to business logic layer [BAL]. The main advantage to use repository pattern is to isolate the data access logic and business logic, so that if changes in any of this logic are made, that cannot effect directly on other logic.

### The Repository and Unit of Work Patterns

The repository and unit of work patterns are intended to create an abstraction layer between the data access layer and the business logic layer of an application. Implementing these patterns can help insulate your application from changes in the data store and can facilitate automated unit testing or test-driven development (TDD).

The CRUD operations using Database-first approach is done by following the below given three steps:

1. Connection to the database
2. Execute the LINQ query on the DbSet<TEntity>
3. Save the changes made in the collection to the database

In the repository of the Data Access Layer, multiple methods are written to perform various CRUD operations over different tables in the database. By using such different methods for each different operation, it is easy to access them whenever required in the business logic. It centralizes data logic or business logic and service logic. It gives a substitution point for the unit tests. It provides a flexible architecture. All the methods required for business logic of Service Incidents Productivity Tracker are written under SIPTRepository in the DATA Access Layer.

**2.6 MVC Implementation**

Service Incident Productivity Tracker is a web application which uses the technology and study of Entity Framework Core and developing the presentation layer using ASP.NET Core MVC. The components in the presentation layer developed by MVC are as follows,

**2.6.1 Model**

A **model** stores data that is retrieved according to the commands from the Controller and displayed in the View.

Model is a collection of classes wherein you will be working with data and business logic. Hence, basically models are business domain-specific containers. It is used to interact with database. It can also be used to manipulate the data to implement the business logic.

**2.6.2 Controllers**

Controllers are essentially the central unit of your ASP.NET MVC application. It is the 1st recipient, which interacts with incoming HTTP Request. So, the controller decides which model will be selected, and then it takes the data from the model and passes the same to the respective view, after that view is rendered. Actually, controllers are controlling the overall flow of the application taking the input and rendering the proper output.

Controllers are C# classes inheriting from System.Web.Mvc.Controller, which is the built-in controller base class. Each public method in a controller is known as an action method, meaning you can invoke it from the Web via some URL to perform an action.

ASP.NET MVC Action Methods are responsible to execute requests and generate responses to it. By default, it generates a response in the form of ActionResult. Actions typically have a one-to-one mapping with user interactions.

For example, enter a URL into the browser, click on any particular link, and submit a form, etc. Each of these user interactions causes a request to be sent to the server. In each case, the URL of the request includes information that the MVC framework uses to invoke an action method. The one restriction on action method is that they have to be instance method, so they cannot be static methods. Also there is no return value restrictions. So you can return the string, integer, etc.

**2.7** **Request Processing**

Actions are the ultimate request destination in an MVC application and it uses the controller base class.

**2.7.1 Routing**

ASP.NET introduced Routing to eliminate needs of mapping each URL with a physical file. Routing enable us to define URL pattern that maps to the request handler. This request handler can be a file or class. In ASP.NET Web form application, request handler is .aspx file and in MVC, it is Controller class and Action method.

Route defines the URL pattern and handler information. All the configured routes of an application stored in RouteTable and will be used by Routing engine to determine appropriate handler class or file for an incoming request.

**2.7.2 The URL Pattern**

The URL pattern is considered only after domain name part in the URL. For example, the URL pattern *"{controller}/{action}/{id}"* would look like localhost:1234/{controller}/{action}/{id}. Anything after "localhost:1234/" would be considered as controller name. The same way, anything after controller name would be considered as action name and then value of id parameter. Routing maps URL to physical file or class.

The request processing is done as follows,

* When a URL arrives, like /Home/index, it is the UrlRoutingModule that inspects and understands that something configured within the routing table knows how to handle that URL.
* The UrlRoutingModule puts together the information we've configured in the routing table and hands over control to the MVC route handler.
* The MVC route handler passes the controller over to the MvcHandler which is an HTTP handler.
* MvcHandler uses a controller factory to instantiate the controller and it knows what controller to instantiate because it looks in the RouteData for that controller value.
* Once the MvcHandler has a controller, the only thing that MvcHandler knows about is IController Interface, so it simply tells the controller to execute.
* When it tells the controller to execute, that's been derived from the MVC's controller base class. The Execute method creates an action invoker and tells that action invoker to go and find a method to invoke, find an action to invoke.
* The action invoker, again, looks in the RouteData and finds that action parameter that's been passed along from the routing engine.

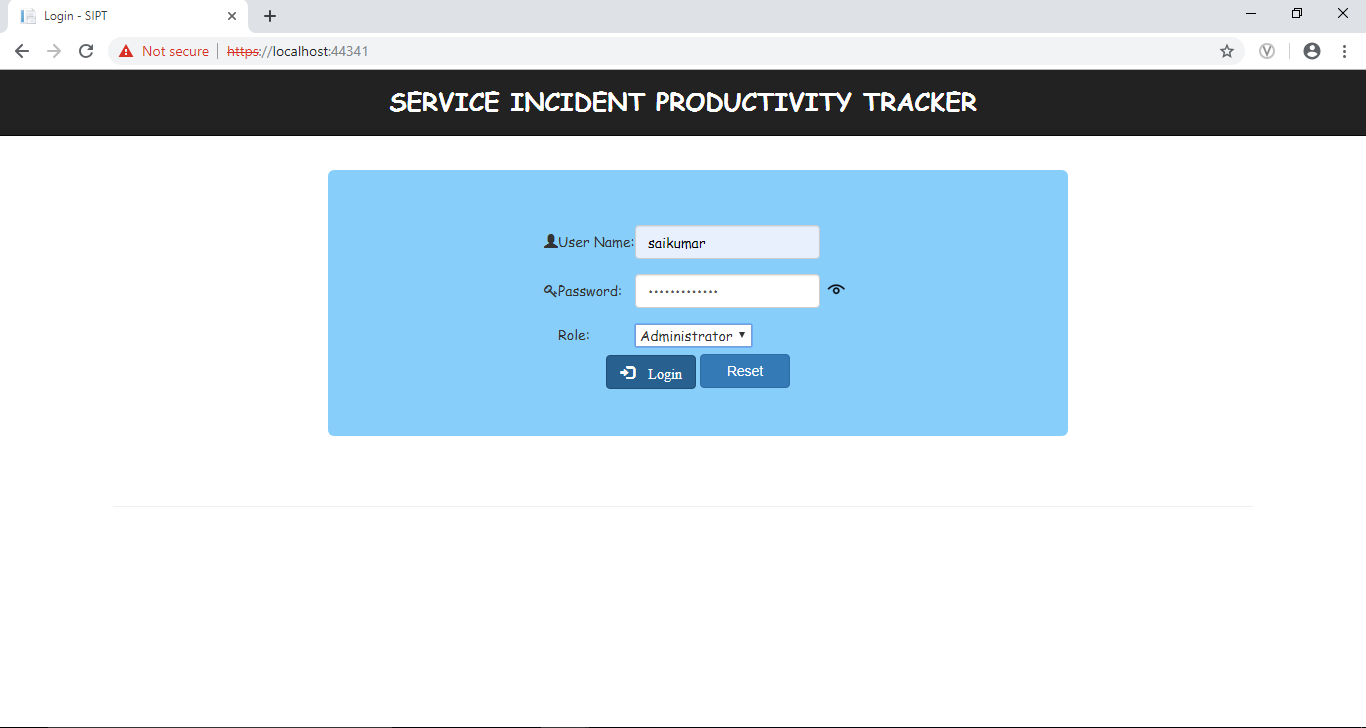
**2.7.3 View**

In an ASP.NET MVC application, there is nothing like a page and it also doesn’t include anything that directly corresponds to a page when you specify a path in URL. The closest thing to a page in an ASP.NET MVC application is known as a **View**.

In ASP.NET MVC application, all incoming browser requests are handled by the controller and these requests are mapped to controller actions. A controller action might return a view or it might also perform some other type of action such as redirecting to another controller action.

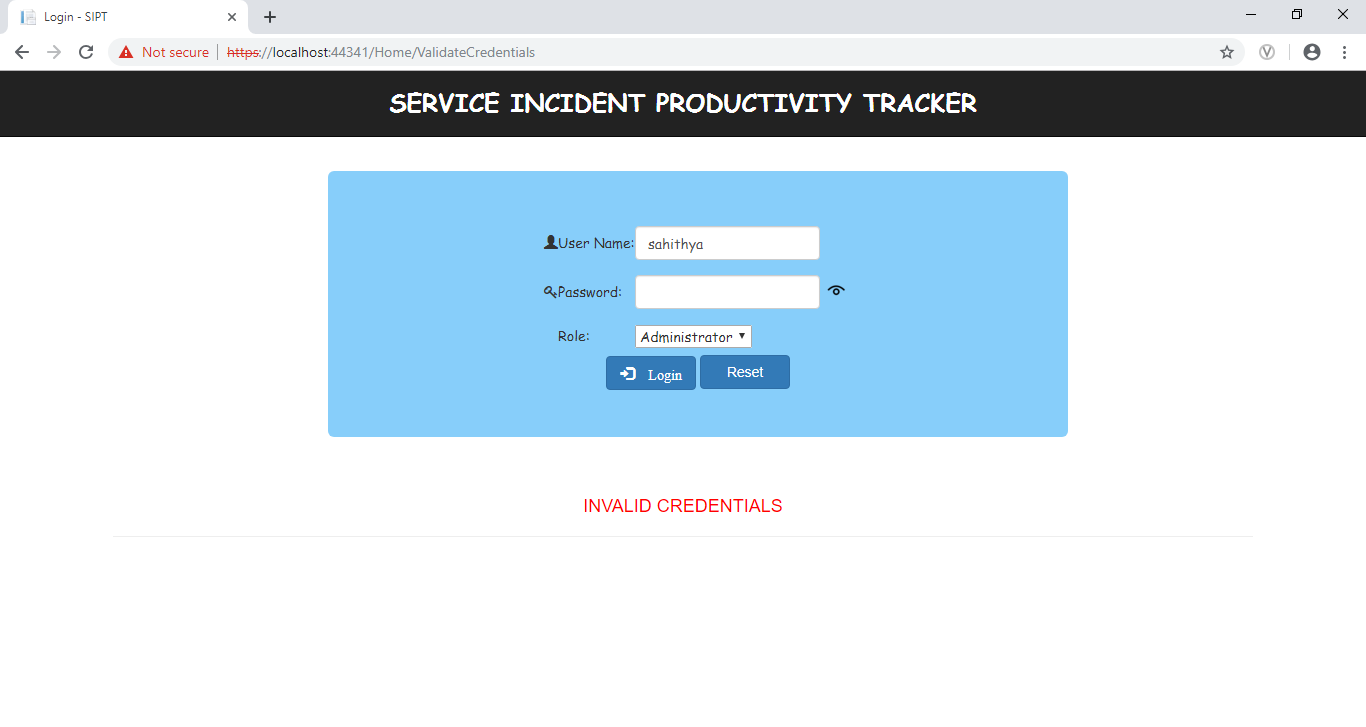
1. **RESULTS**

Service Incident Productivity Tracker is a web application which needs validation of the user to use it. So, the first page loaded when the project is run is the Login Page, which requires the User Name and Password of the user.



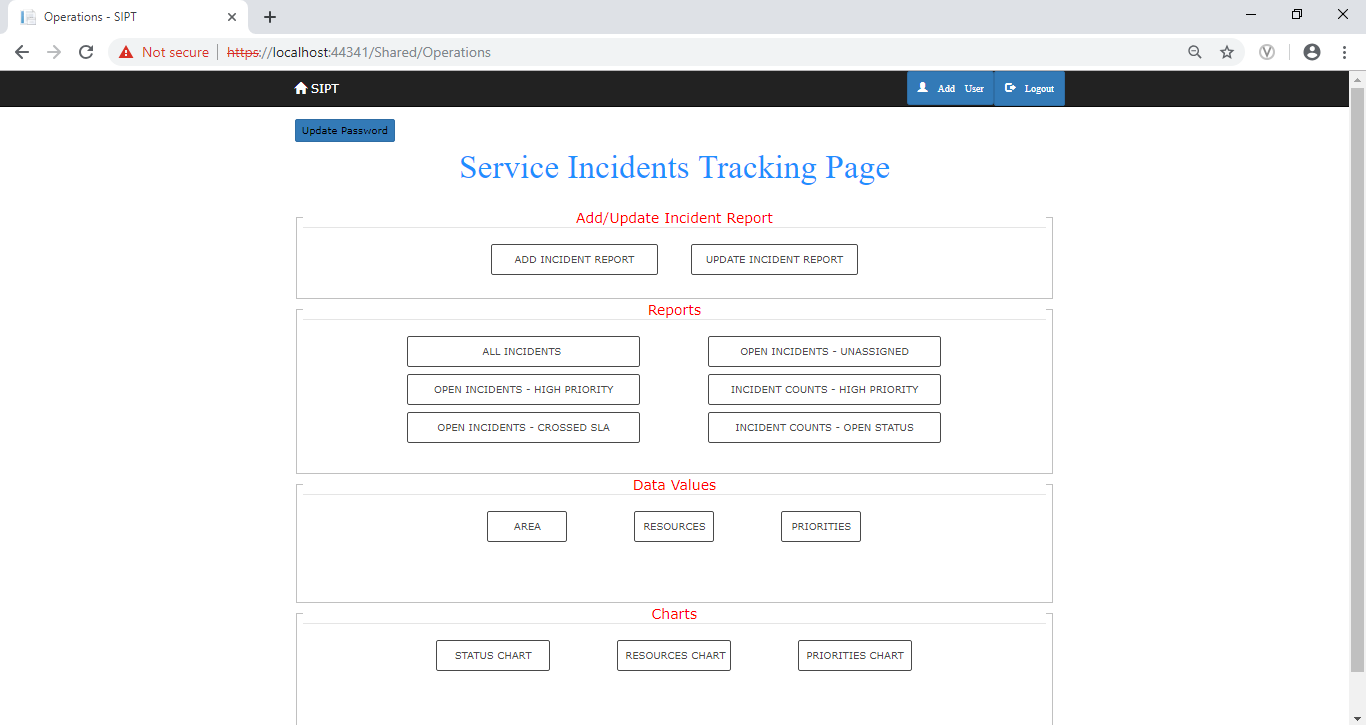
**Figure 3.1**: Login Page of administrator

If the user enters any invalid detail, he/she is not allowed to make use of the Service Incident Productivity Tracker. Instead they are shown an error message “INVALID CREDENTIALS”.



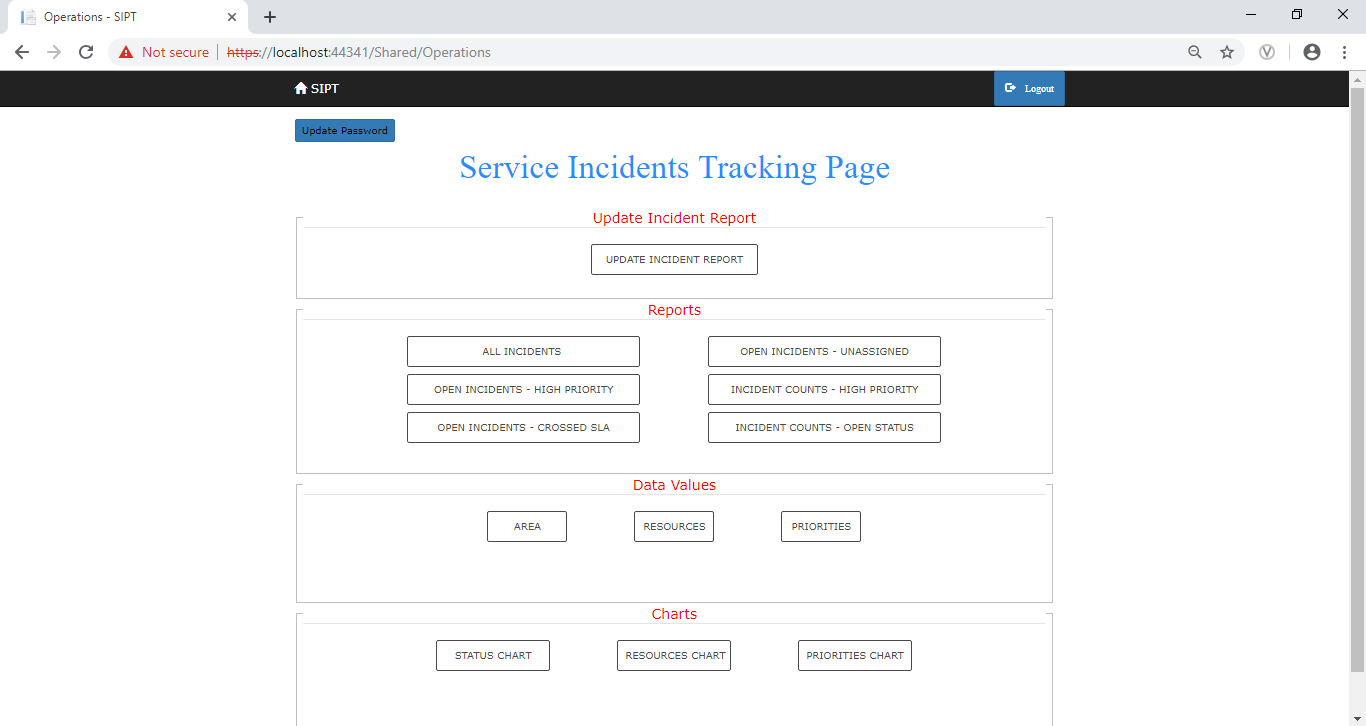
**Figure 3.2**: Login Page of Administrator with invalid credentials

Consider that the user enters valid credentials and the user is an Administrator, they are redirected to the Operations page which allows the user to perform various operations. As the person logged in is an administrator they can both Add and Update an incident. They can also download the reports of incidents with high priority and incidents with target date crossed and many others.



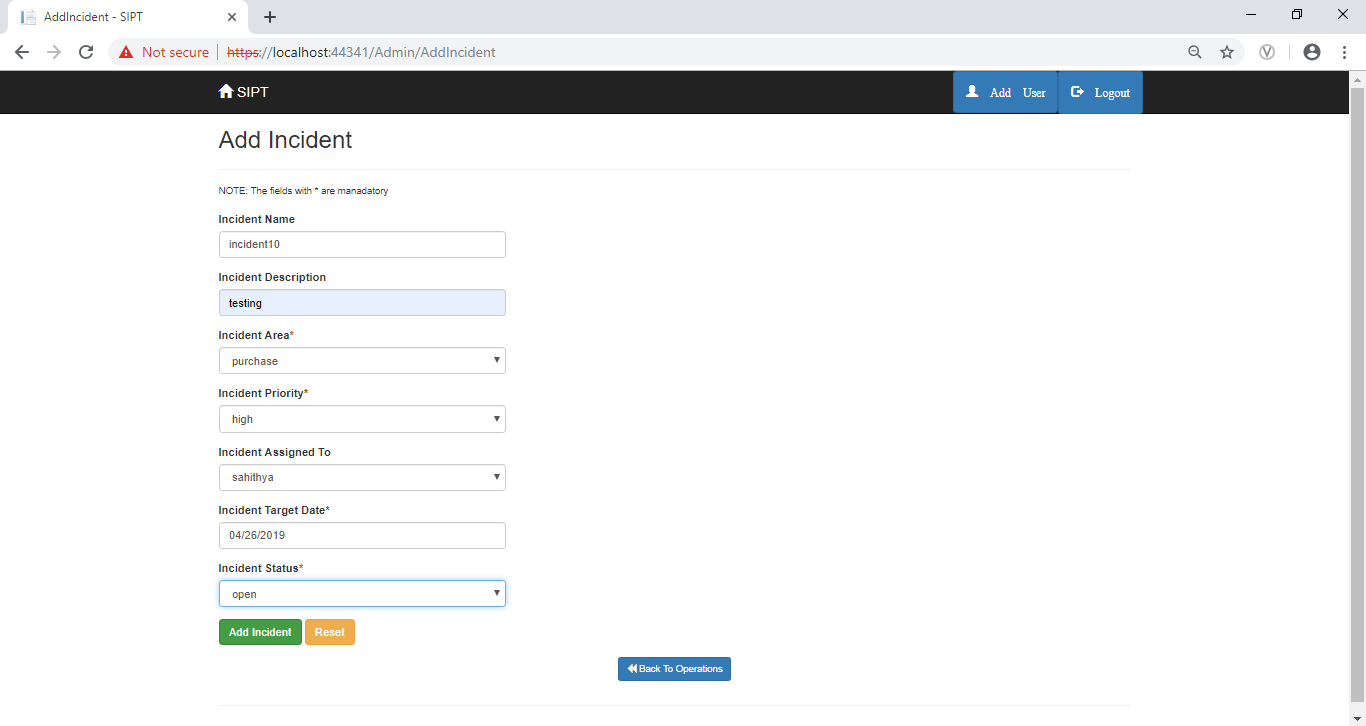
**Figure 3.3**: Incident Tracking Page of Administrator

Considering that the person logged in is a user, they are redirected to the Operation page with only Update Incident as they are not given any permissions to manipulate the database by adding new incident. All the other operations are similar to that of the administrator.



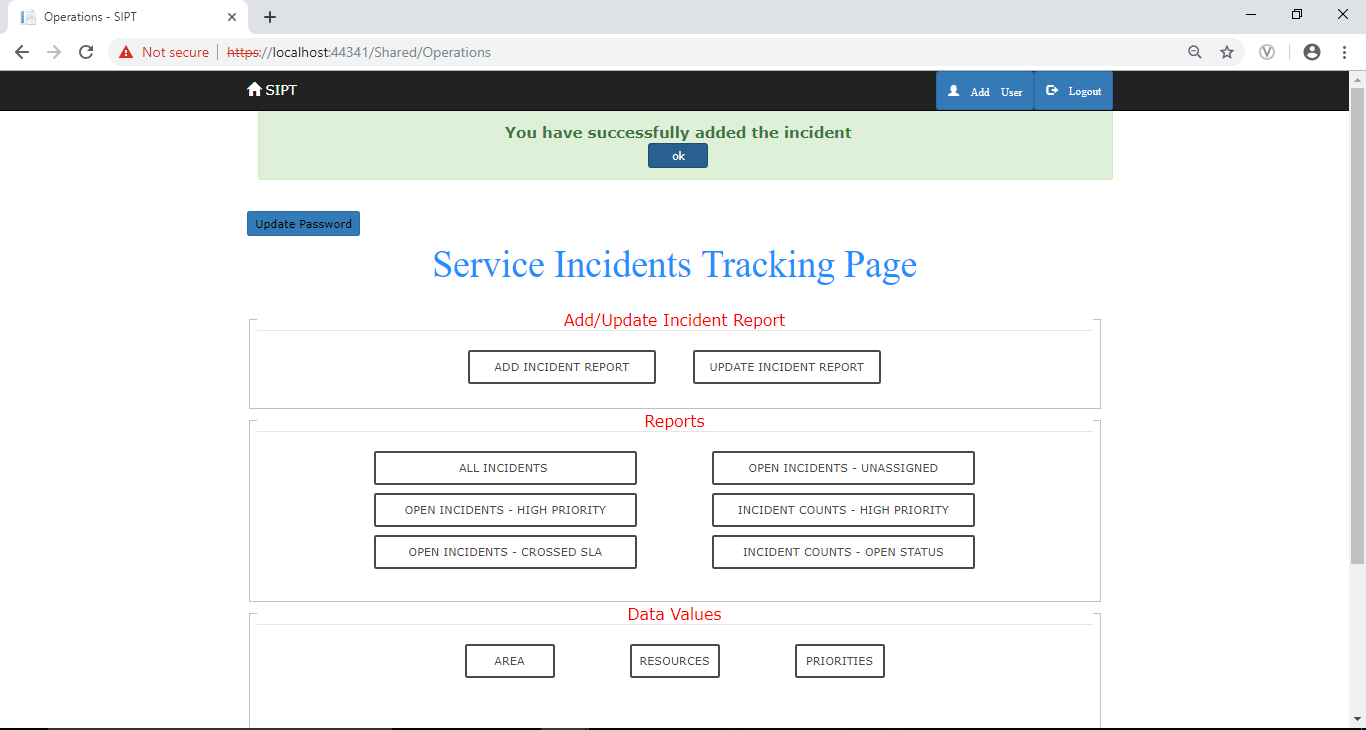
**Figure 3.4**: Incident Tracking Page of User

Add Incident Report button enabled for Administrator allows to add new Incident in to the incident table of database where the user enter details like Incident name, Incident Description, and selects Incident Area, Incident Priority, Incident Assigned to, Incident Target date and Incident status.

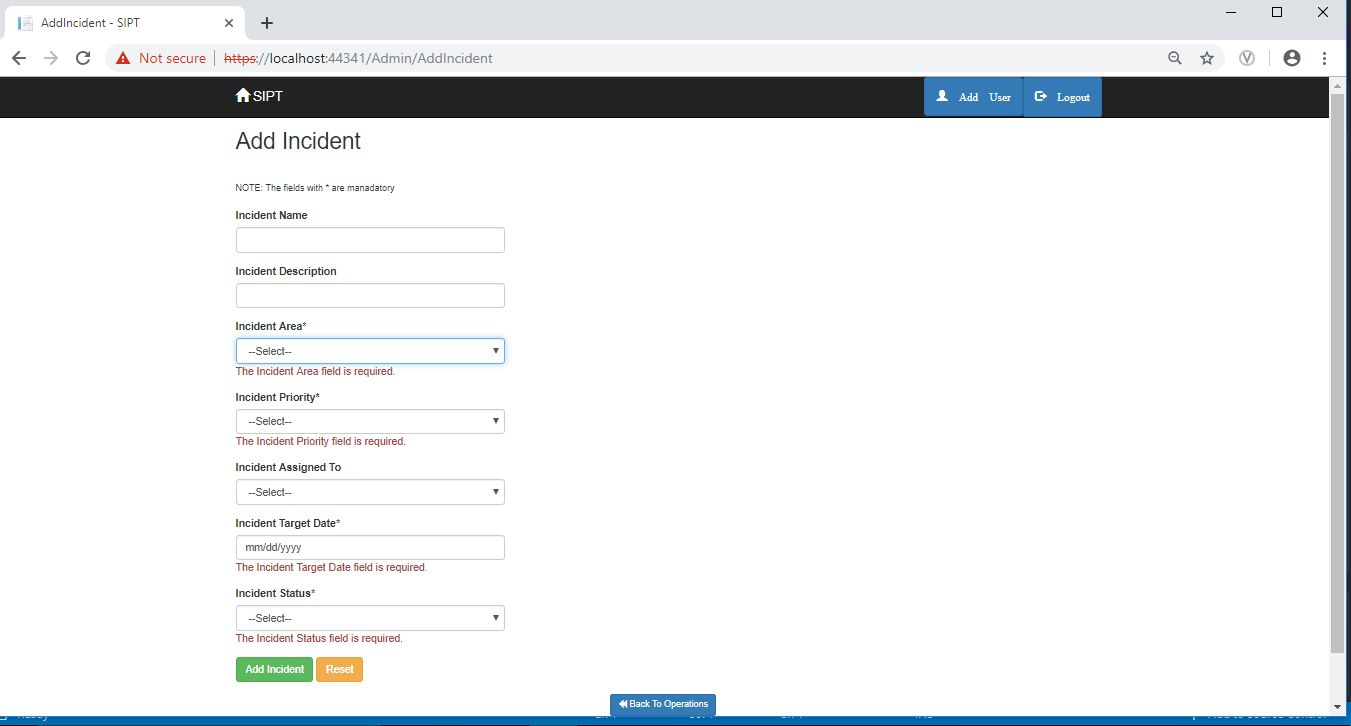


**Figure 3.5**: Add Incident Page of Administrator

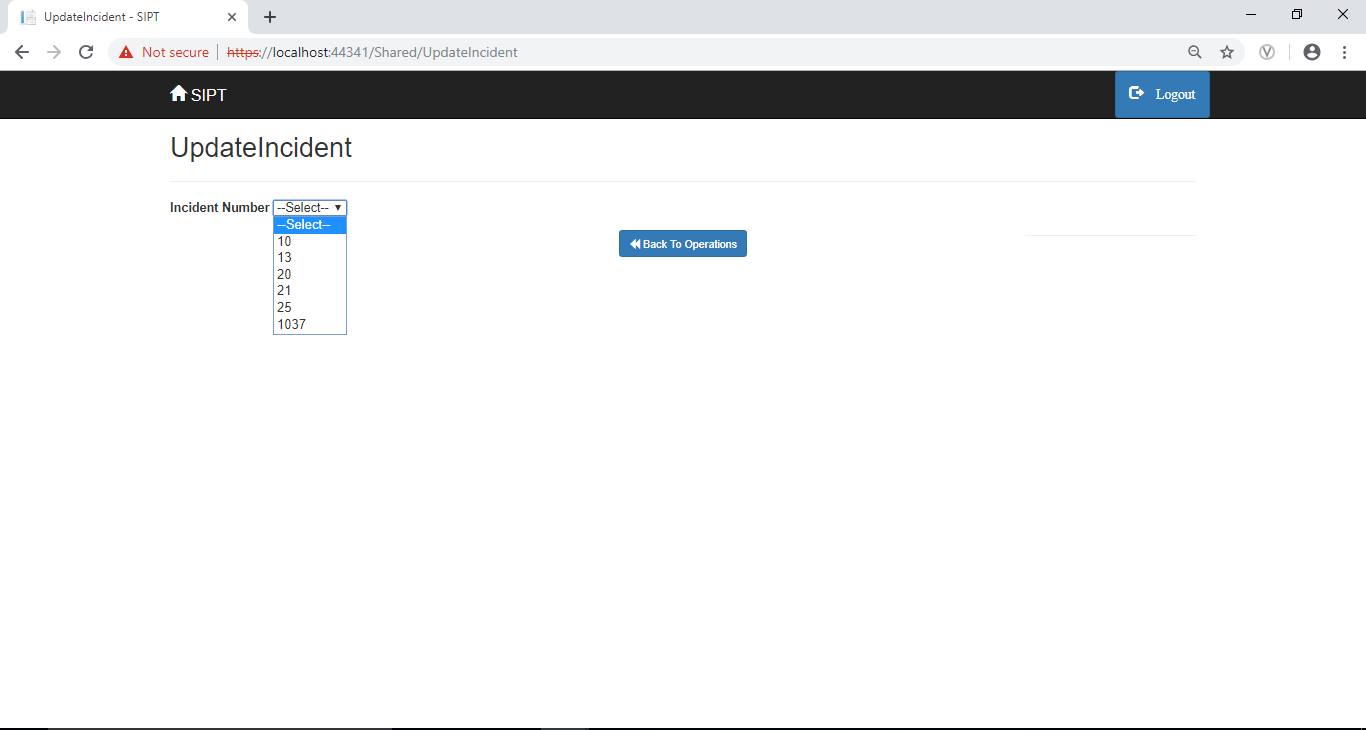
By clicking on Add Incident, if the details are valid then a Notification is displayed on Operations page saying “You have successfully added an incident”.



**Figure 3.6**: Add Incident Page of Success

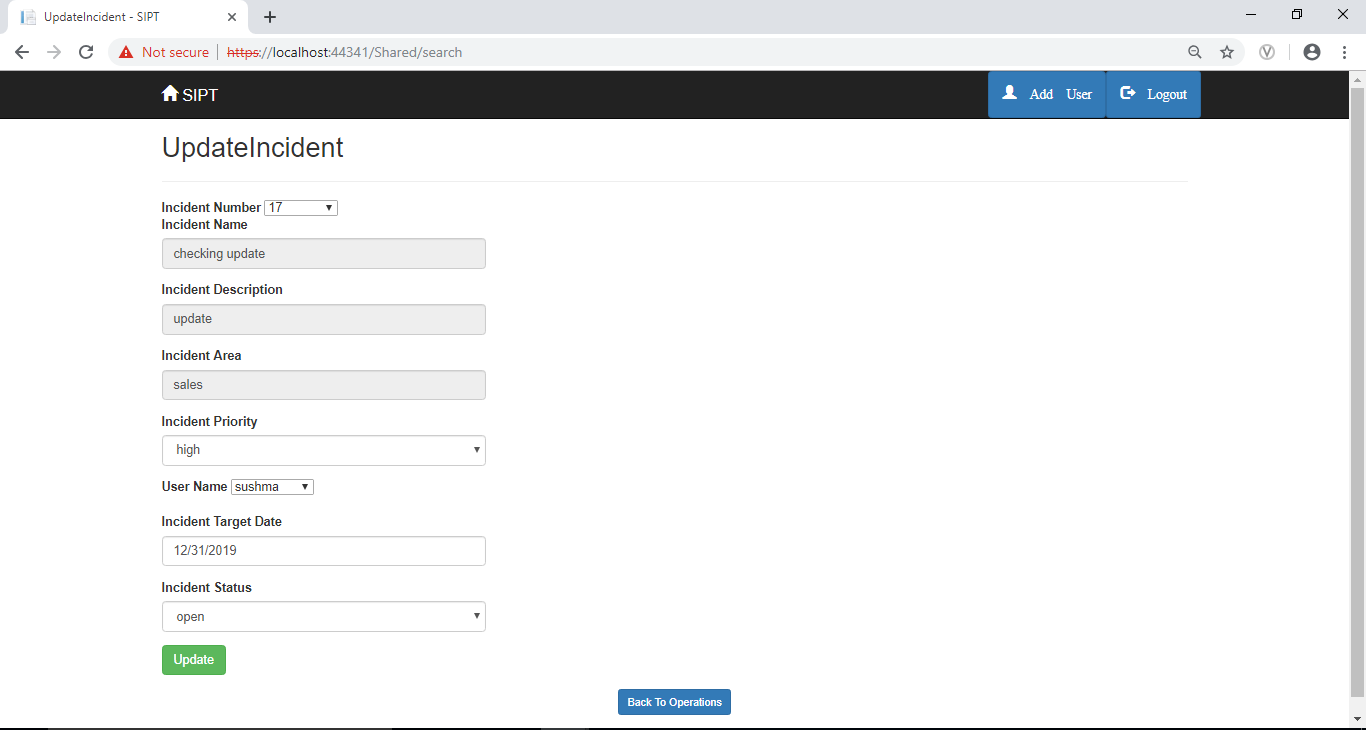
If the User did not select any of the required fields and click on Add Incident Button, then error message is shown to that particular field as “required field” as shown below. Reset button to Reset values in all the fields. 

**Figure 3.7**: Add Incident Page of User Invalid Credentials

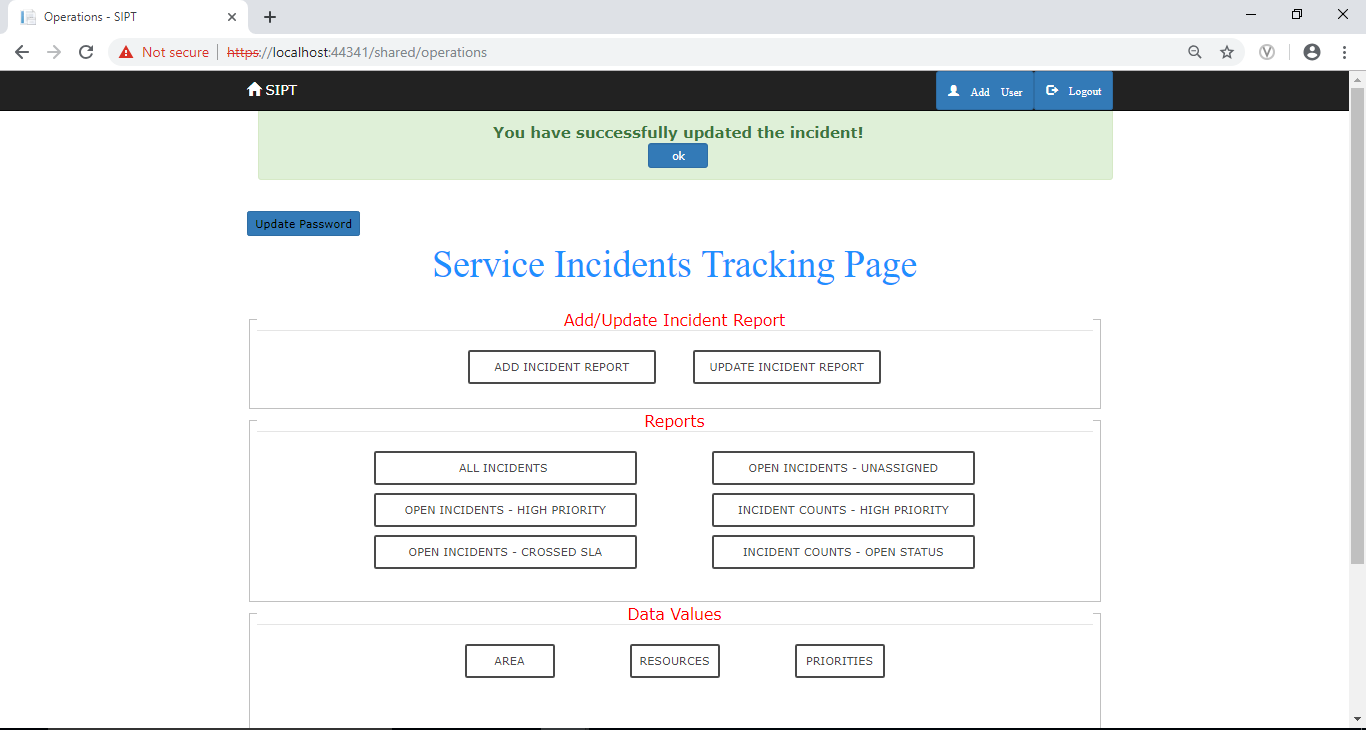
Update Incident Button allows the user to update the incidents allocated only to that particular user and keep the information up to date. If the user is an administrator then he/she is allowed to update all the incidents present in Database.

**Figure 3.8**: Update Incident Page of User

Once the user selects the incident required from the drop down list, all the fields and their values of the selected incident are shown as follows and allow user to modify Incident Priority, User to whom it is assigned, Incident Target date and Incident Status.

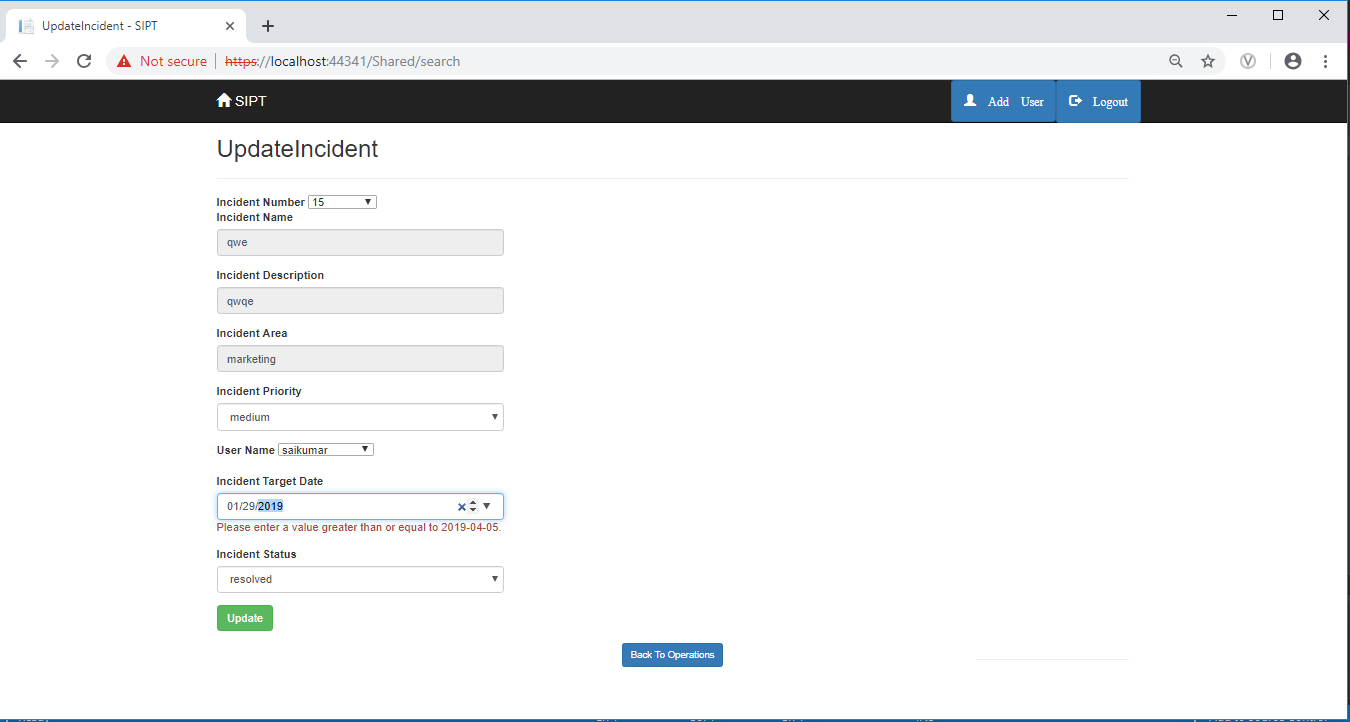


**Figure 3.9**: Update Incident Page of User

If the User clicks on the update button present in the Update page, then a notification is displayed on operations page saying “You have Successfully updated the Incident”. 

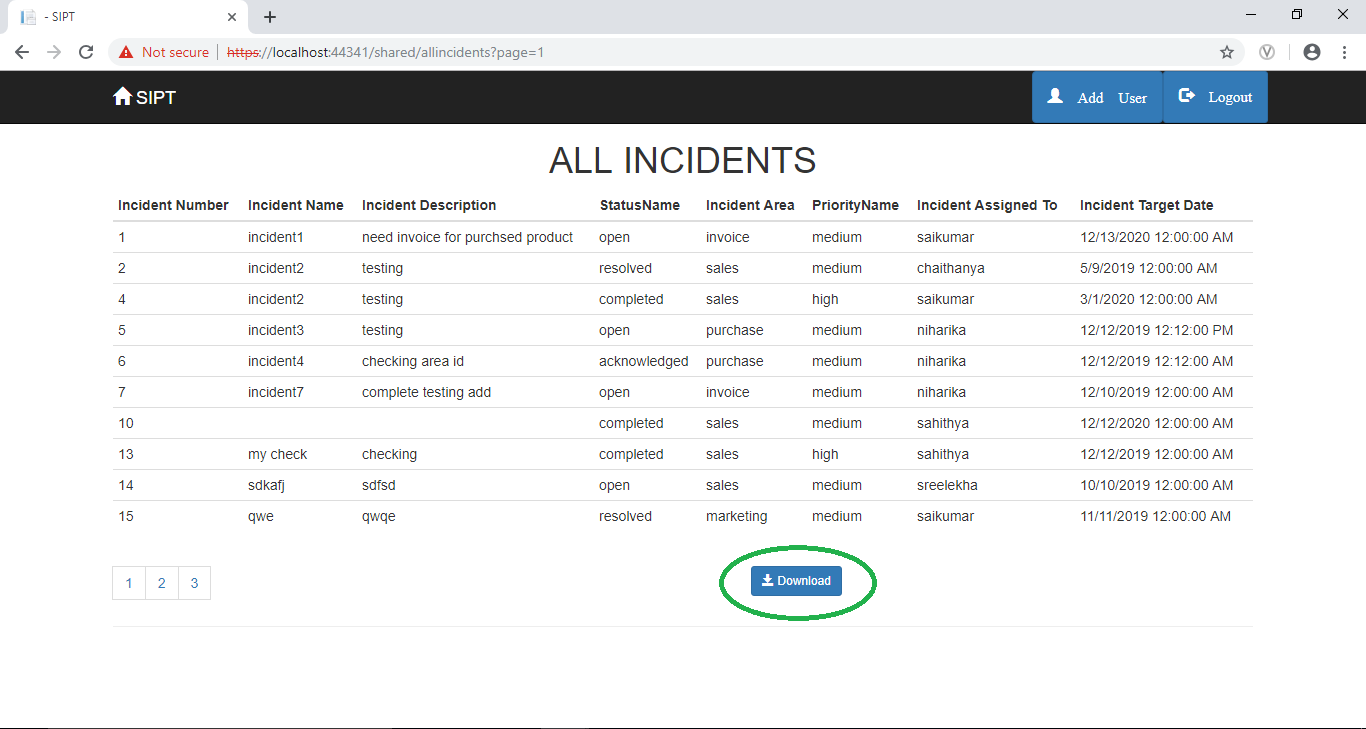
**Figure 3.10**: Update Incident Page of Success

If the Target date selected in update page is not greater than the current date, then it shows an error message in the page dynamically.



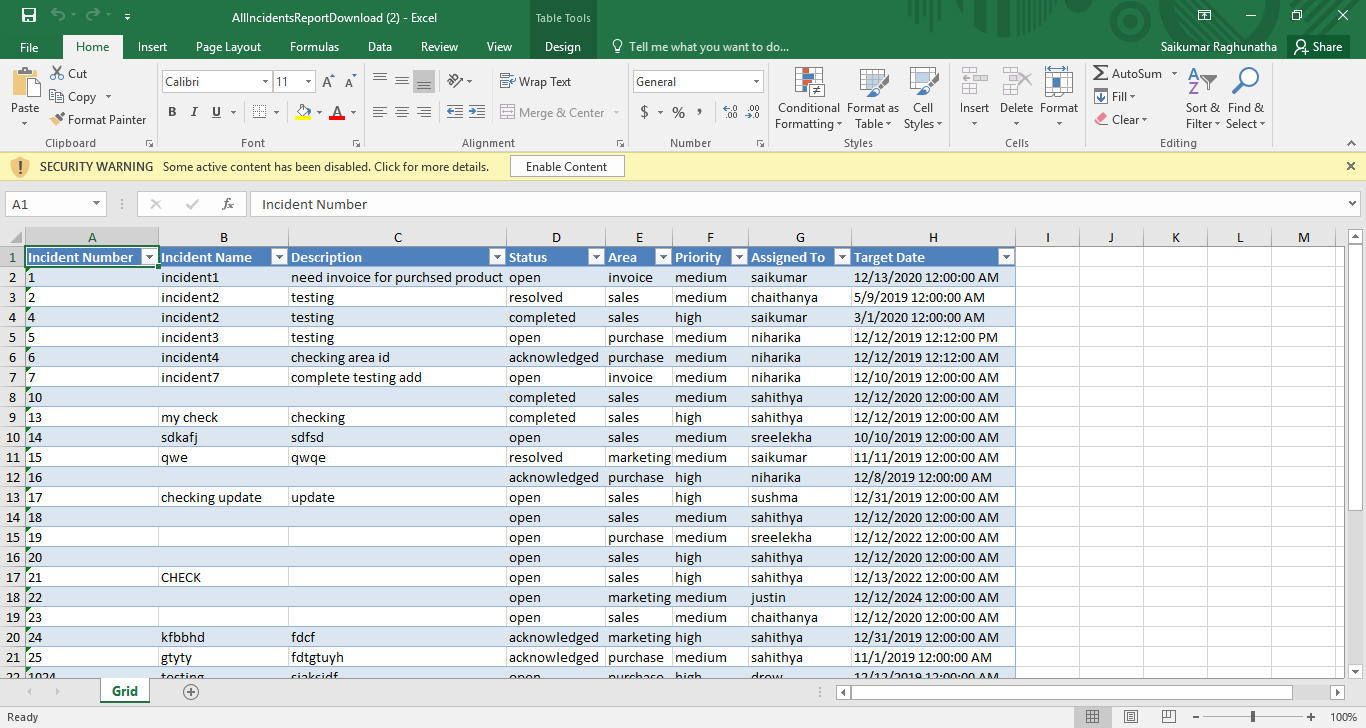
**Figure 3.11**: Update Incident Page on Invalid Date

Once the person logged in, clicks on the button **All Incidents** under **Reports** section, they are redirected to the All Incidents page with the download button which will help the user to directly download the data in an excel sheet.

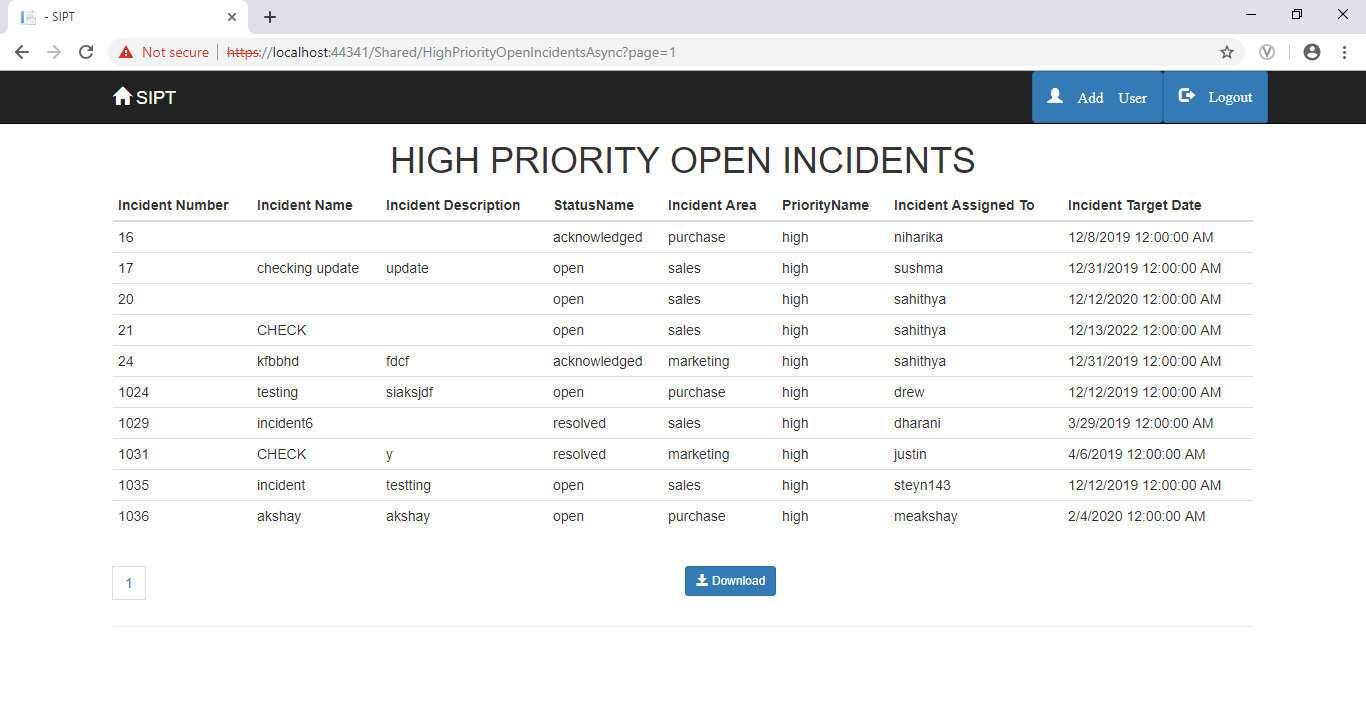


**Figure 3.12**: All Incidents page

The downloaded excel sheet will look as follows,



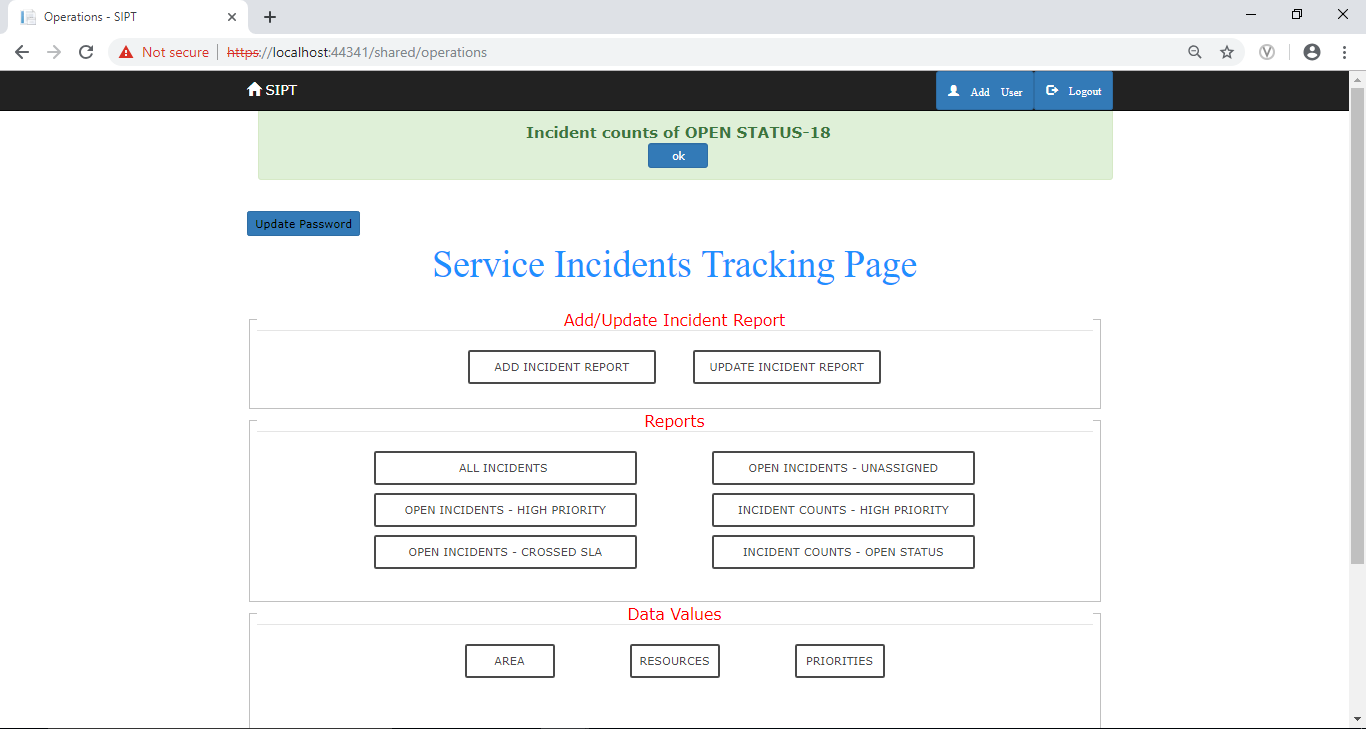
**Figure 3.13**: All Incidents Page of Excel sheet

Once the person logged in, clicks on the button **Open Incidents – High Priority** under **Reports** section, they are redirected to the Open Incidents High Priority page with the download button which will help the user to directly download the data in that page in an excel sheet.

**Figure 3.14**: High priority Open Incident Page

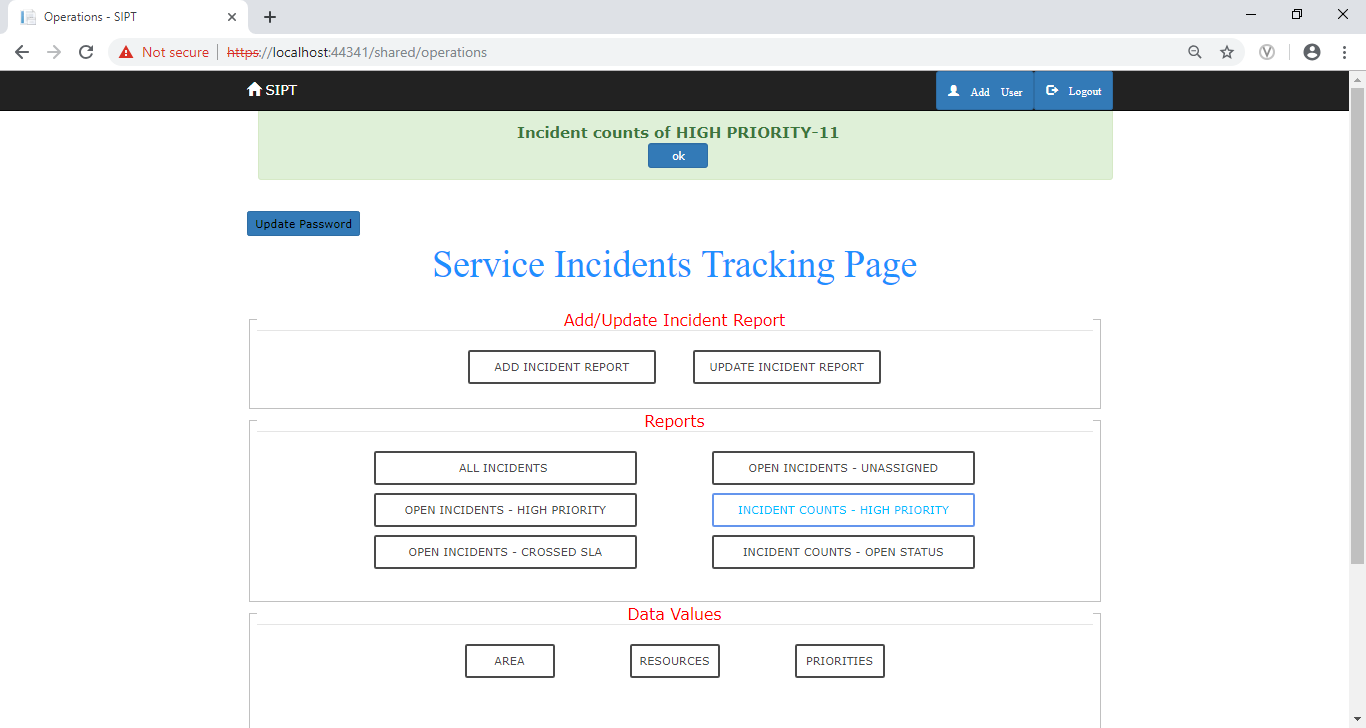
Similarly, all the other buttons have the functionality where the user gets redirected to the corresponding pages with the functionality to download and save the data in that particular page in an excel sheet.

When the user clicks on Incident Count- Open Status button a notification is displayed and displays the count of incidents which has the status as Open.



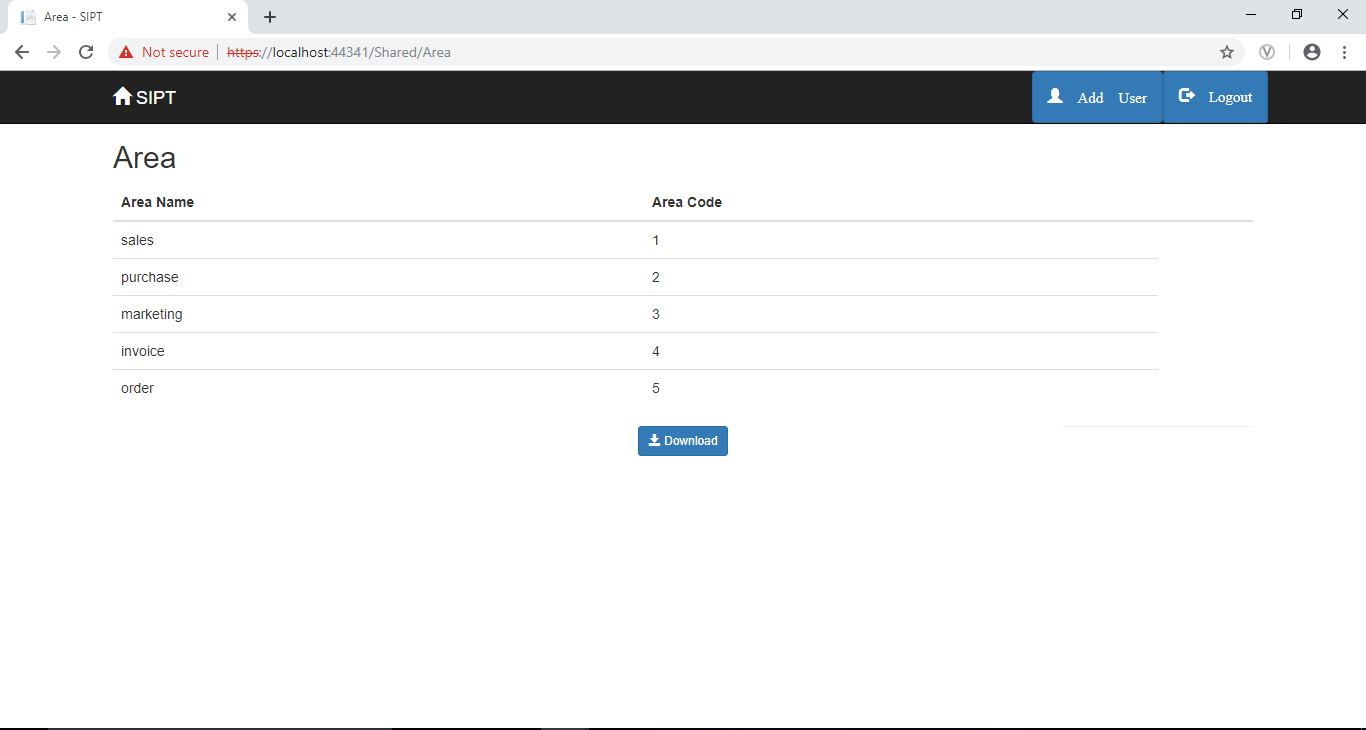
**Figure 3.15**: Incident Count of Open Status

When the user clicks on Incident Count- High Priority button a notification is displayed and displays the count of incidents which has the Priority as High.



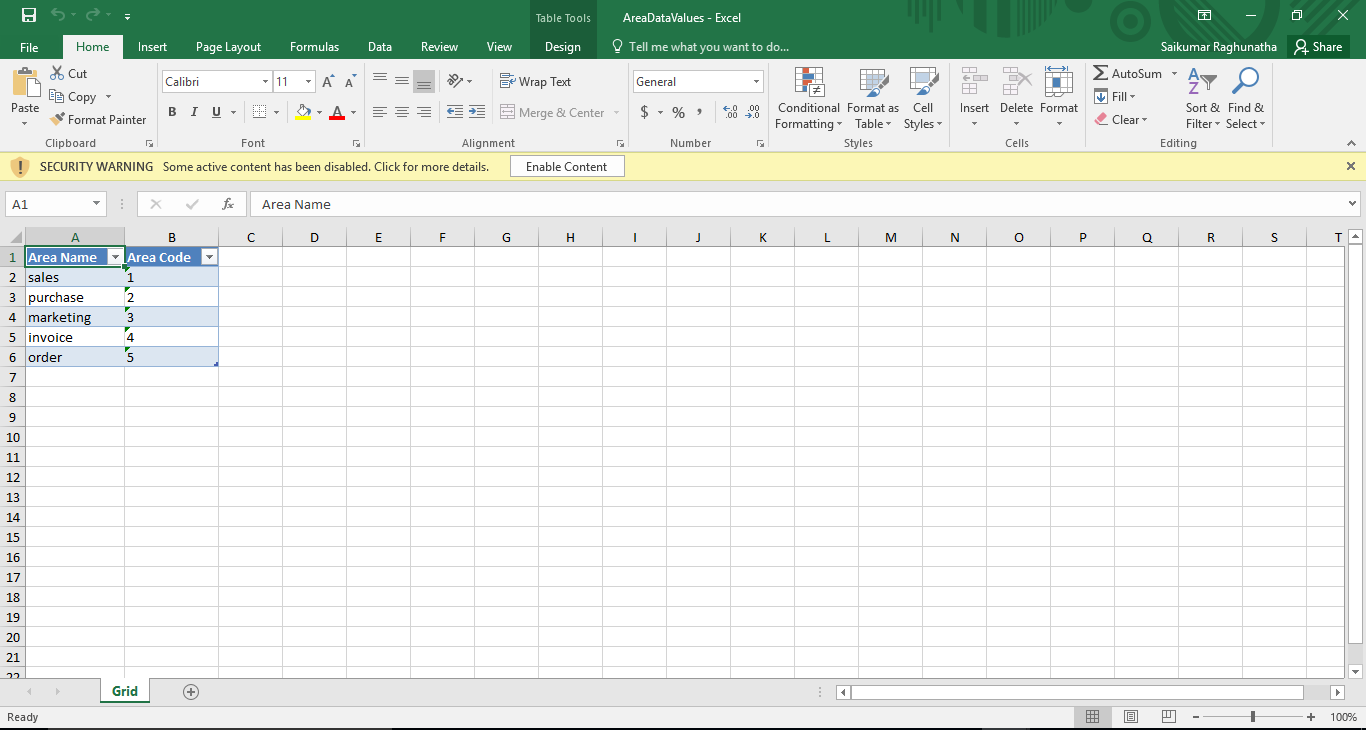
**Figure 3.16**: Incidents Count of High Priority

Under the **Data Values** section, there are different categories of how the data in the database can be divided into, like **Area, Resources** and **Priorities**. The Areas is the category by which the incidents are classified based on the area of work. The resources are the details of the employees who work on their assigned incidents. Priorities refer to the level of importance the incidents are given. All these values are shown in a page once selected, with the functionality of downloading them in an excel sheet.



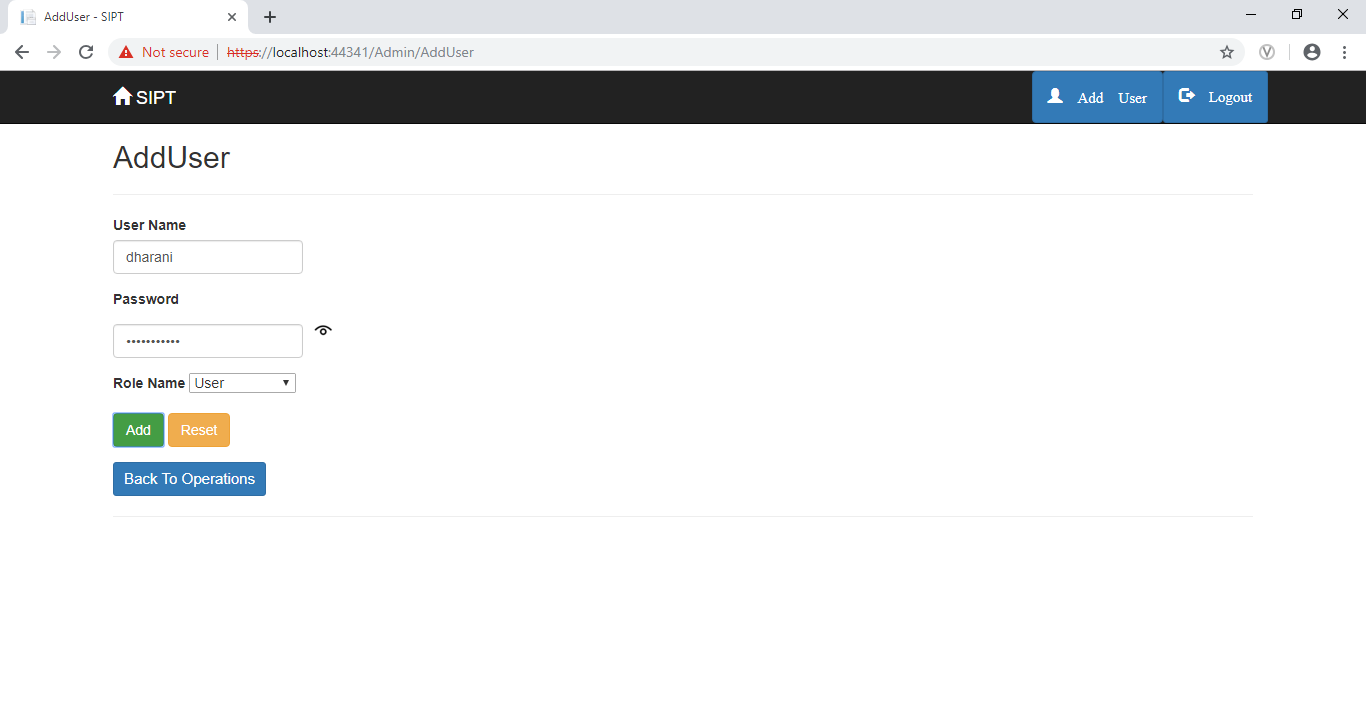
**Figure 3.17**: Area Page of Incidents

The downloaded excel sheet will look as follows,

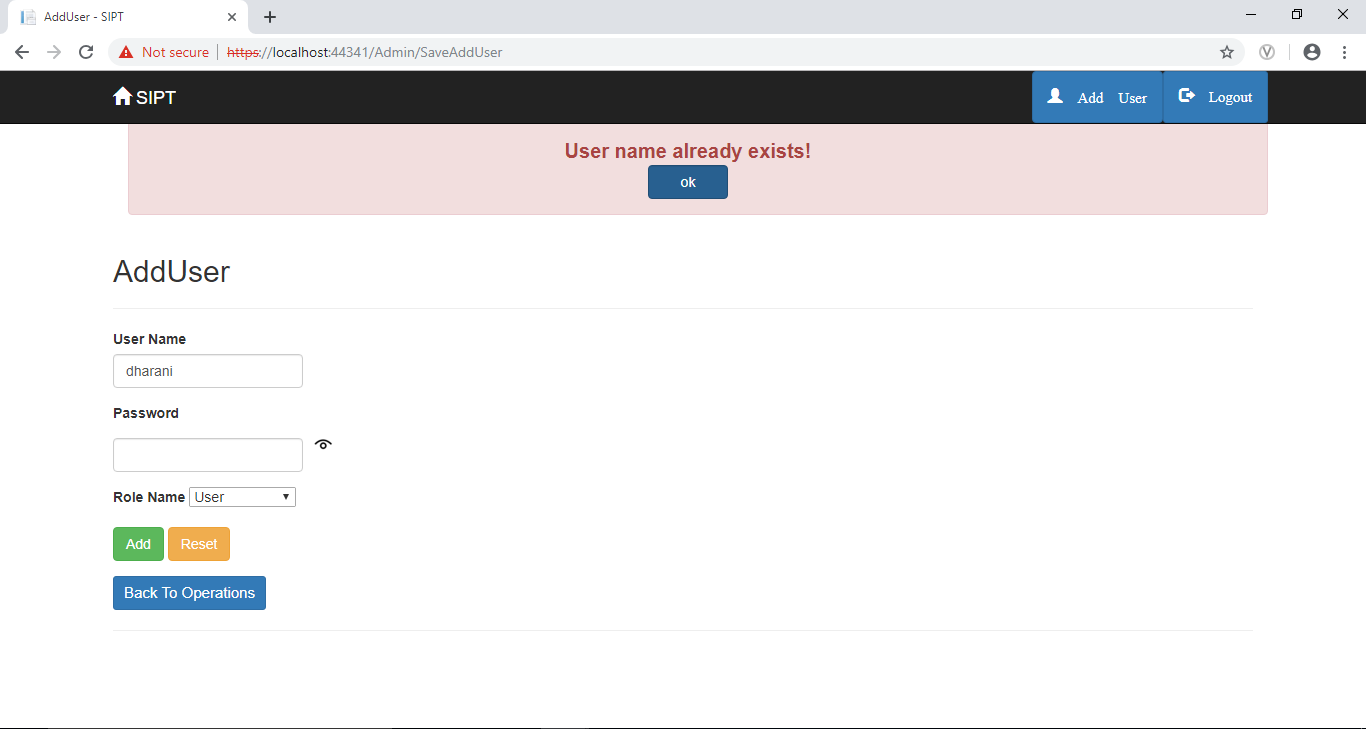


**Figure 3.18**: Areas Page of Excel Sheet

The Administrator is given some special privileges like Adding a User. If the administrator wishes to add a new user, then they select the Add User button. On clicking it, the administrator is redirected to add user page where they assign the new user a username and a dummy password. Once the administrator hits the add user button, the new user is added to the User table in the database.

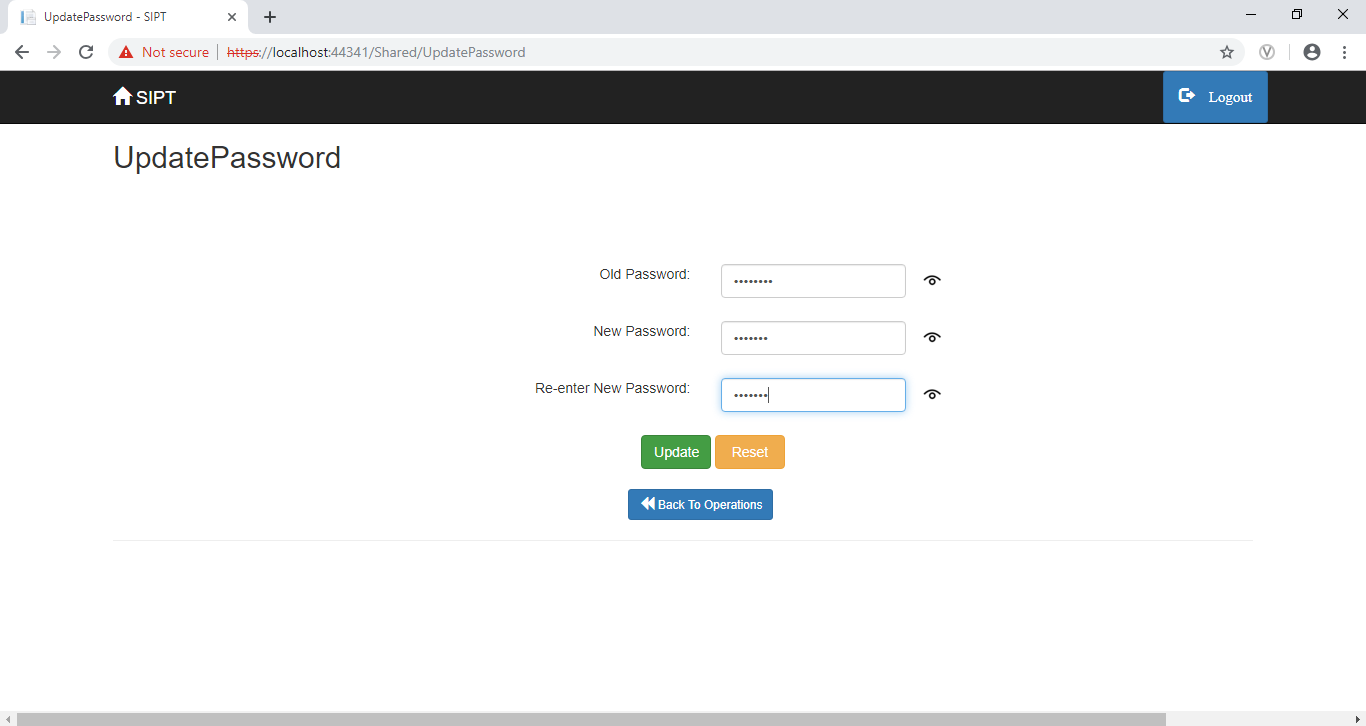


**Figure 3.19**: Add User Page

If the User enters a username that is already present in the database, it is not possible to add a new user, at that instant an error message is shown. 

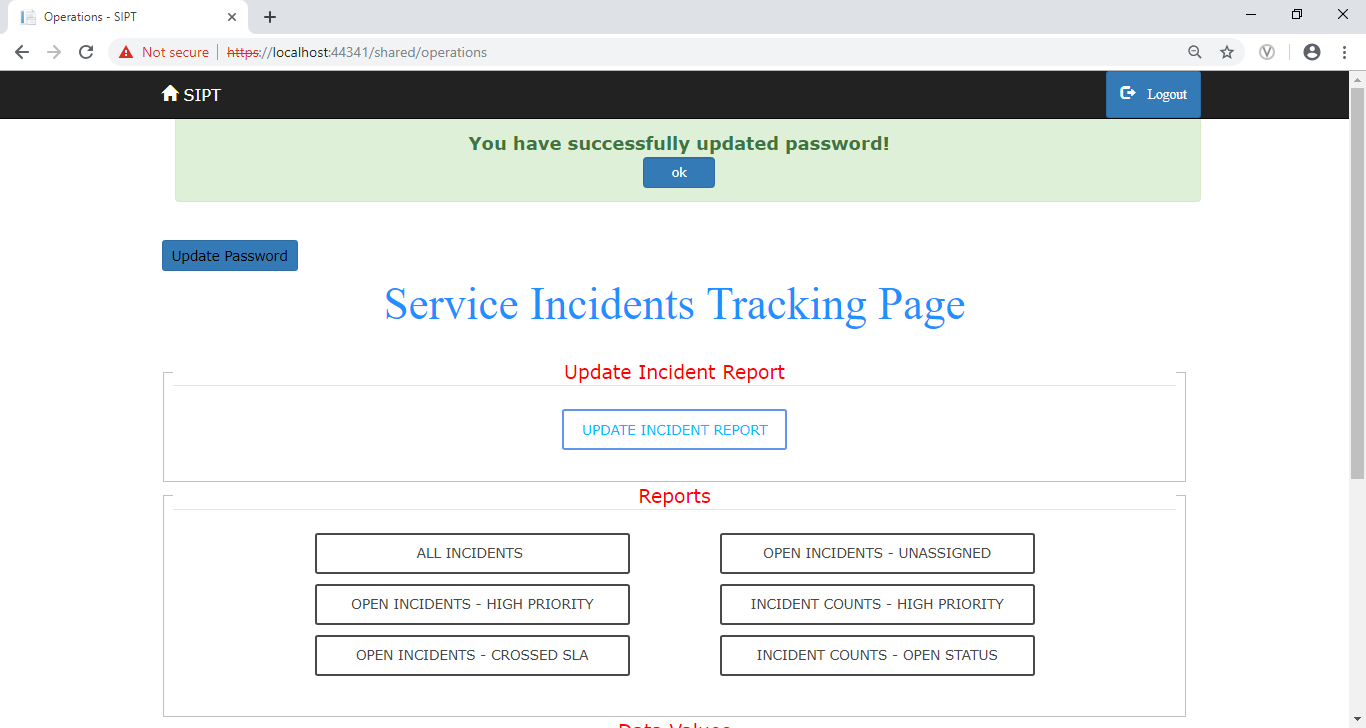
**Figure 3.20**: Add User page showing User Name already Exists

The user needs to have a strong password. A password is considered Strong if it is a combination of Upper Case letter, Lower Case letter, a digit and alphanumeric characters. It should have a minimum length of 8 characters and a maximum of 16 characters. The user must have to change the password either at the beginning from the dummy password assigned by the administrator to user own password and also in regular intervals. This is done by the Update Password button. The User is provided with this functionality to keep their password updated. Once the user clicks on Update Password button it is redirected to updatepassword page. Once the user hits Update button, the password is updated in the database and the user can continue with the new password.



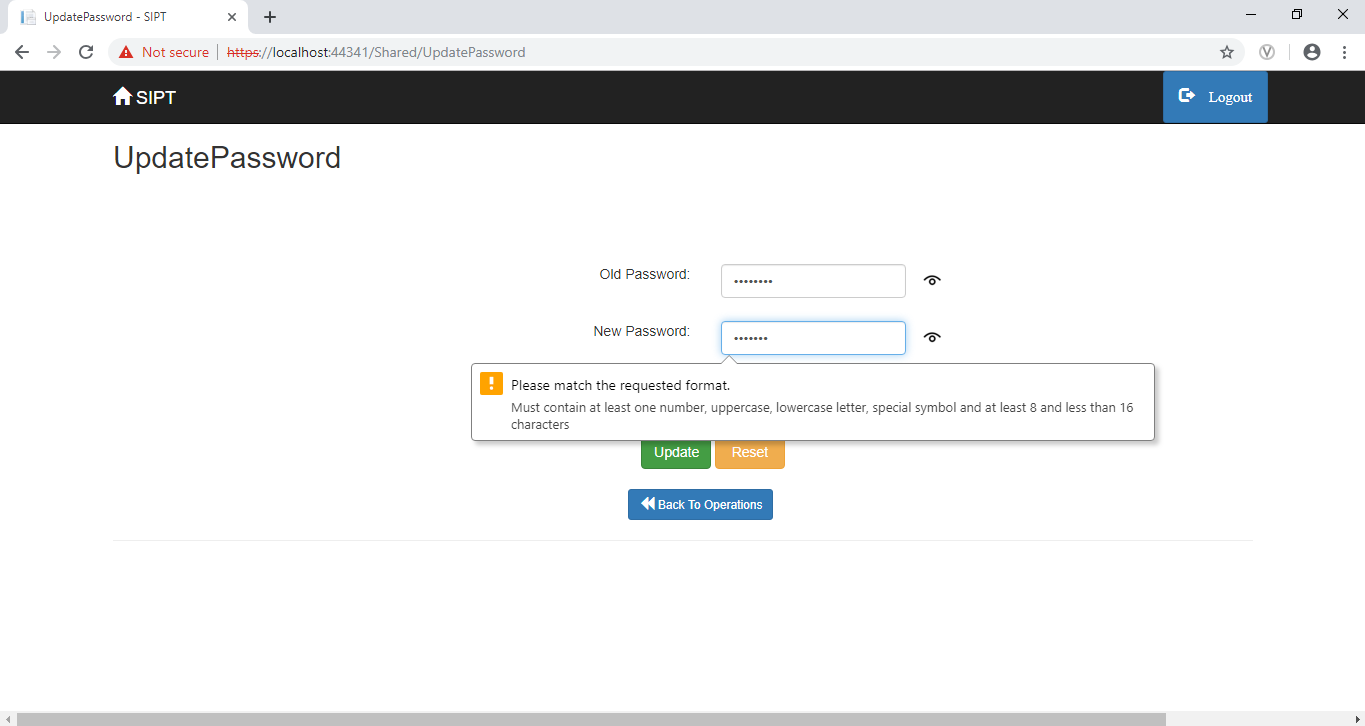
**Figure 3.21**: Update Password Page

Once the password is updated, a notification is shown with the message “Password is Updates successfully”.



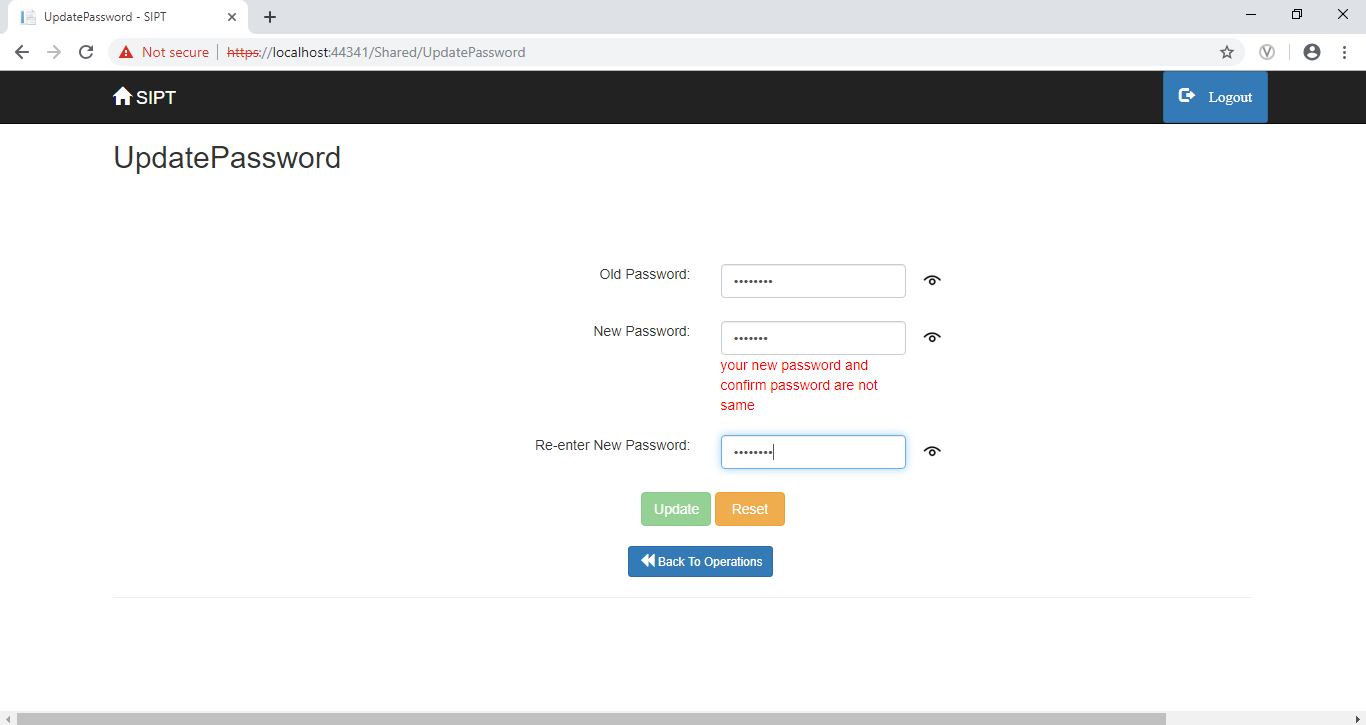
**Figure 3.22**: Successful updation of password

If the new password is not valid, then it throws a warning message showing the requirements of password.



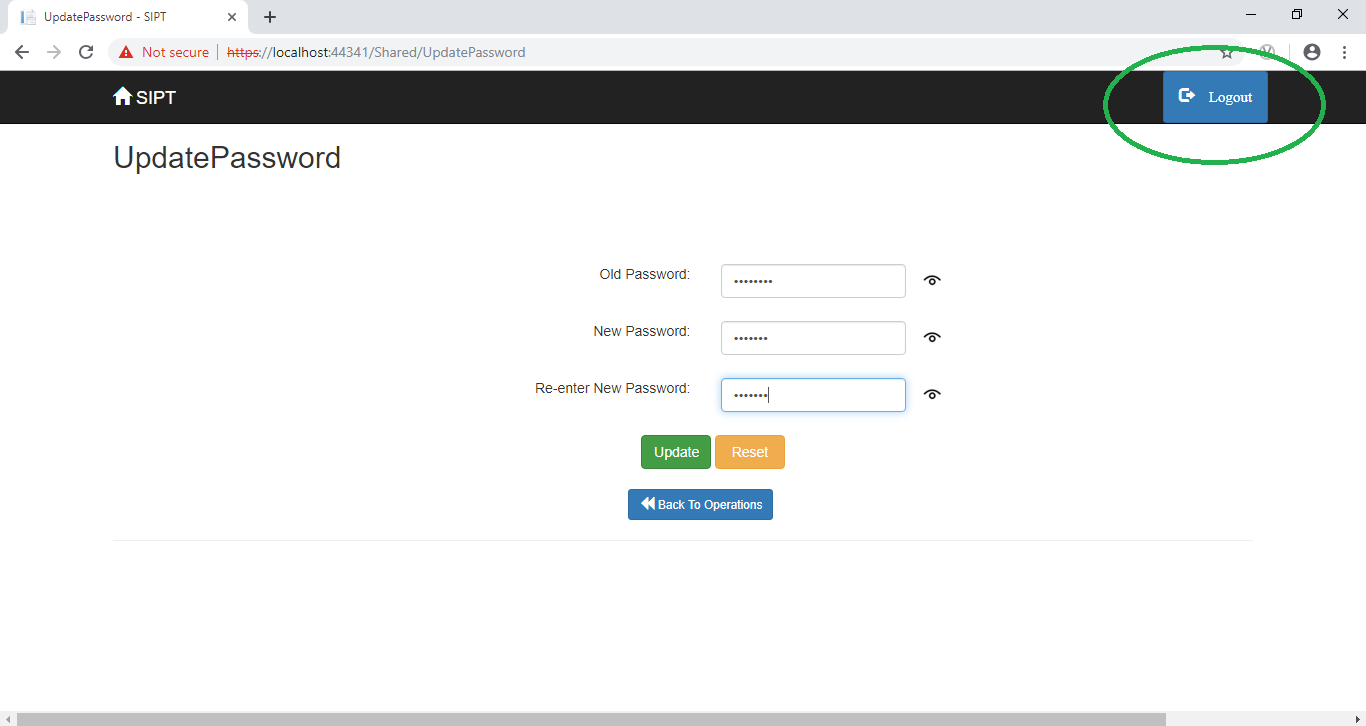
**Figure 3.23**: Page showing password is invalid

If the Re-enter password field does not match with new password, then it throws an error message showing that the “your new password and confirm password are not the same”.



**Figure 3.24**: New password and Old password Mismatch

Once the user is done with the application then they can Logout from the session by hitting the Logout button. Once the Logout button is hit, it redirects the user or the administrator back to the login page.



**Figure 3.25**: Logout Page for User

###### CONCLUSION

Service Incident Productivity Tracker resulted in improving overall productivity with better fleet management that in turn offers better return on your investments. Better scheduling or route planning can enable you handle larger jobs loads within a particular time. Service Incident Productivity Tracker in the business world improves communication medium, performance monitoring and increases productivity. So in the corporate world, it is going to play a major role. This system has many advantages such as Easy Accessibility, Effective Communication Strategy, automated as possible and makes the access of incidents much more convenient.

The Service Incident Productivity Tracker delivers following business benefits

* Smooth business operations.
* Improved productivity.
* Satisfied end users.
* Maintaining consistent service levels.
* Proactive identification and prevention of major incidents

Upgrading this setup is very easy which makes it open to future a requirement which also makes it more efficient.