ASD ESSENTIAL EIGHT

**FRONT-END AND BACK-END DOCUMENTATION**

ASD ESSENTIAL EIGHT CYBER MITIGATION TOOLKIT ASD ESSENTIAL EIGHT SIT764 CLASS OF T3 2020

# 1.Introduction

The ASD Essential Eight forms the baseline of the proposed product; Cyber Security Capability and Maturity Consultancy Services, SecureBiz. The aim of this proposal is to “Research, assess and determine key applications of the Australian Signals Directorate (ASD) Essential Eight mitigation strategies in the small to medium enterprise business environment”. The project involves designing the Frontend, Backend, Database and questionnaire design for all mitigation strategies.

Backend development refers to server-side development. It involves activities such as designing database and schema, scripting and designing architecture. Front end involves creating a GUI content with which user can interact. In short, Front end refers to the user interface while the backend refers to the server, the web application and the database that works together to process and deliver data to front end. Both front end and back end part of application needs to work efficiently to produce a Minimum viable product (MVP). There are various challenges associated with front end activities such as User Experience (UX) design, mock-ups, prototyping and user interface development. The backend to deliver required content requested by the user. Integrating front end and back end involves depicting how the content is generated in front end, flow of requests to back end, how back end handles and processes requests and finally, how responses are sent back to Front end. These are discussed in detail in this document.

## 1.1 Major Purpose of Document:

The purpose of the Front-End and Back-End documentation is to provide current and future users of the SecureBiz product with the technical aspect of the product. This document also includes information regarding the application security features implemented to secure the website. Backend services are consumed by front end interfaces and applications. Although backend is hidden from the client and managed only by the developer, it is essential to document the features and services to track the overall progress of backend. This document serves to simplify complexities in understanding integration of front end with backend, current state of project and interactions between various modules used to design backend. This includes the structural diagram representing the entire website, navigation flow featuring how the front-end works, Database Design, Flow of request by User from Front-end to Back-End, processing of the request in Backend and flow of response/result back to Front-End

## 1.2 Management of this Document:

## This document references specific features of the product under development subjected to change over time. This document, therefore, should be upkept with the relevant system changes introduced each sprint. Changes, including additions and modifications to this document should be tracked by a log in the table 0.1: Contributors.

|  |  |  |
| --- | --- | --- |
| Date | Contributor | Summary of Contribution |
| 10/09/2020 | Chahat Choudhary | Started preparing the documentation. Inserted all the information related to system architecture and software design, Introduction |
| 12/09/2020 | Chahat Choudhary | Included the information regarding the web application Overview along with inserted the working screen shots of the website |
| 14/09/2020 | Chahat Choudhary | Preparing the flow diagram linking back-end and front-end, navigation flow Diagrams explaining the functioning of the website |
| 15/09/2020 | Chahat Choudhary | Preparing the flow diagram linking back-end and front-end, Process flow Diagrams explaining the functioning of the backend in calculation of the logic and result display |
| 16/09/2020 | Chahat Choudhary | Designing the sub document- Admin portal and explaining about it here |
| 17/09/2020 | Chahat Choudhary | Designing the sub document- Report page and pdf and explaining about it here |
| 13/02/2021 | Hugo Ng | Updated sections relevant to T3 2021 frontend development. |

# 2.System Architecture and Software Design:

The below figure shows the architecture of web application. It defines the interaction between applications, middleware and database-technology used in designing them, flow of requests/response over the entire application. When user requests a particular page, it is sent through the web hosting server (API), which acts as an intermediator, and transmits it to server.  The backend server conforms to model view controller (MVC). Type of response is based upon the type of incoming request.

If user requested for static content such as HTML pages of the website, it is directly sent from Files repository of the application. The static content created using files of CSS, JavaScript, Images and other files. If user requested for dynamic content, it is first processed by server, some server side scripts are run, the request is processed and the demanded page is displayed in front of the user which has been designed using CSS, JavaScript, HTML, ReactJS and then when there is need of response or calculation based on some logic it is also processed in the server and is sent back to Client browser.

A close up of a map

Description automatically generated

## 3.Web Application Overview

The Landing page or the home page consists of options like:

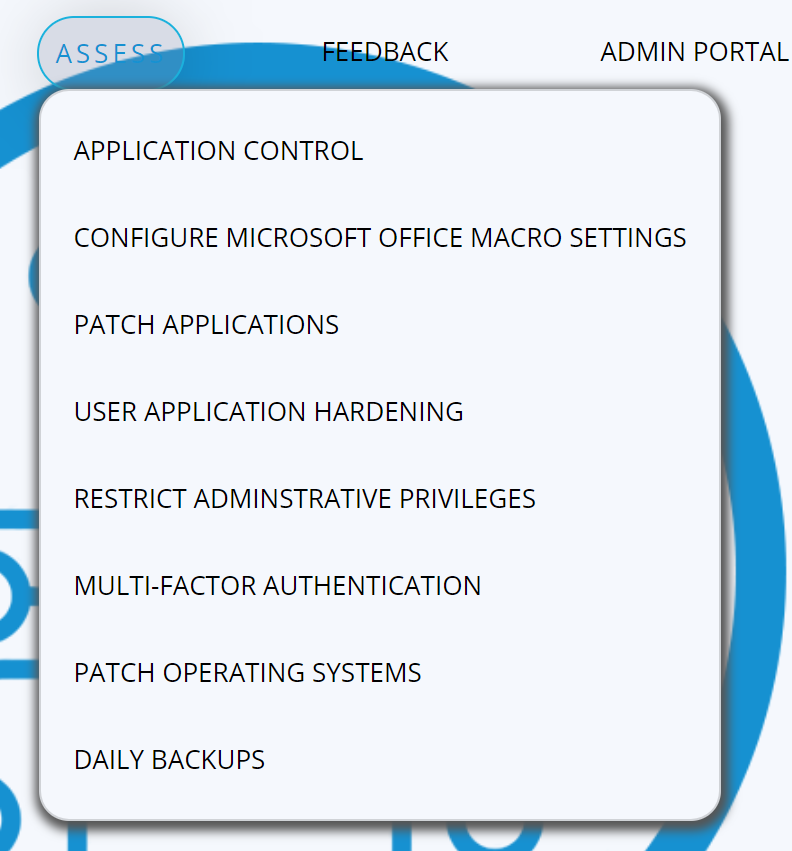
1. **The “Get Started” button:**

When the user first accesses the web application, they are presented with a prominent button labelled “Get Started” on the left panel of the screen. When the user selects this button, a survey launches which iterates through each of the core questions of each of the mitigation strategies. At the end of the survey, A table is presented to the user which list all the maturity level results for the user. User has option to view a short report is which describes the outcome of the survey including the maturity level reached for each of the mitigation strategies along with the recommendations. User also has option to download the PDF file listing the maturity level for each mitigation strategy along with the mitigations/recommendations.

2. **Assess tab** - Mitigation Strategy Breakdown and Analysis:

The “Assess” option in the top-of-page toolbar contains a clickable list of each of the eight mitigation strategies. Selection of one of these mitigation strategies launches a survey which iterates through each of the core and advanced questions related to that mitigation strategy. At the end of the survey, a short report is produced which describes the outcome of the survey including the maturity level reached for that mitigation strategy, and recommendations to improve related areas of the client’s security posture. The user has option to download the report in the form of PDF describing all the elements present in the report.

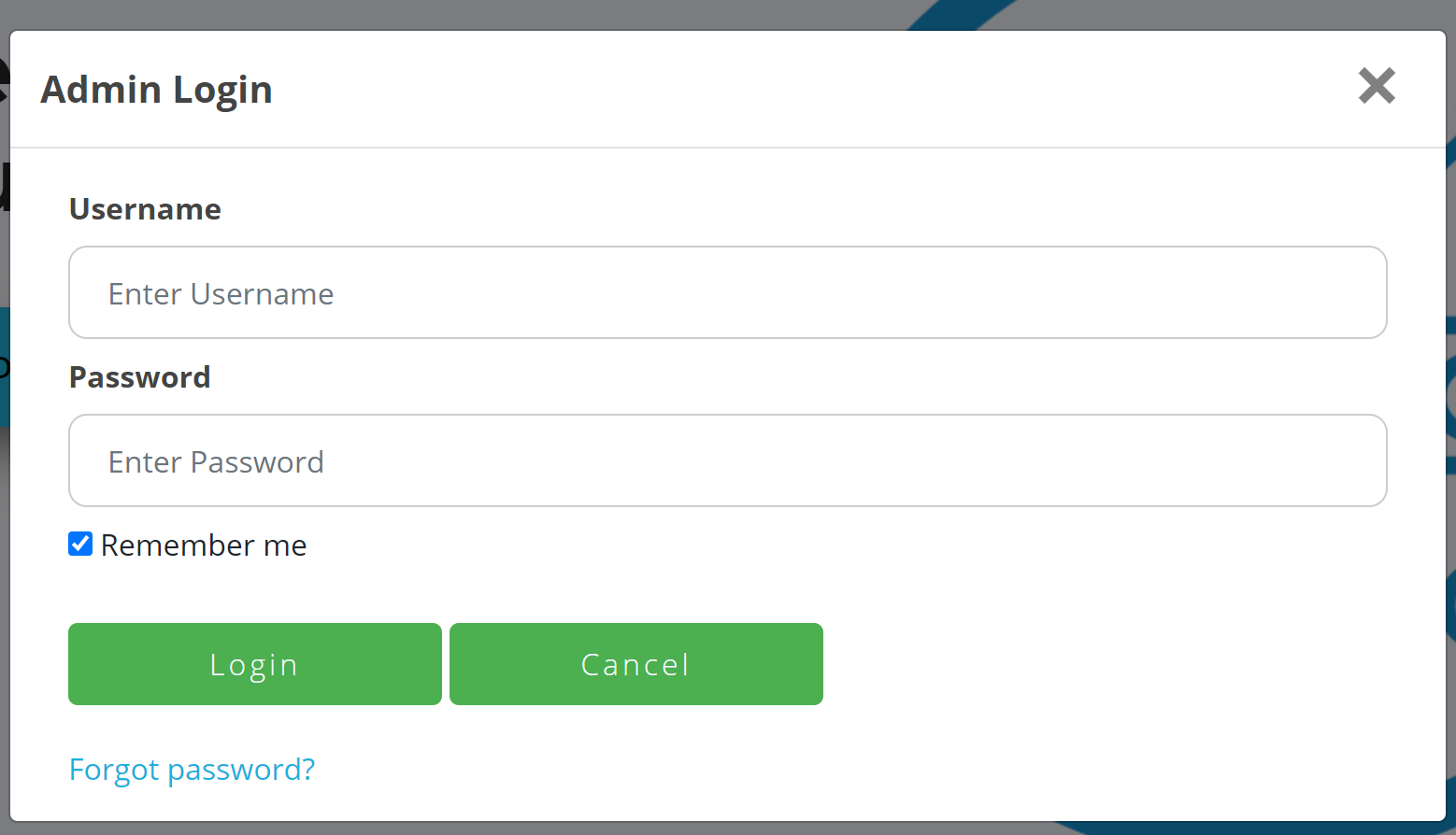
The below screenshot shows how the option will appear to the user when they click on the Assess Tab.



3. **Admin Portal** (Only for admin Users):

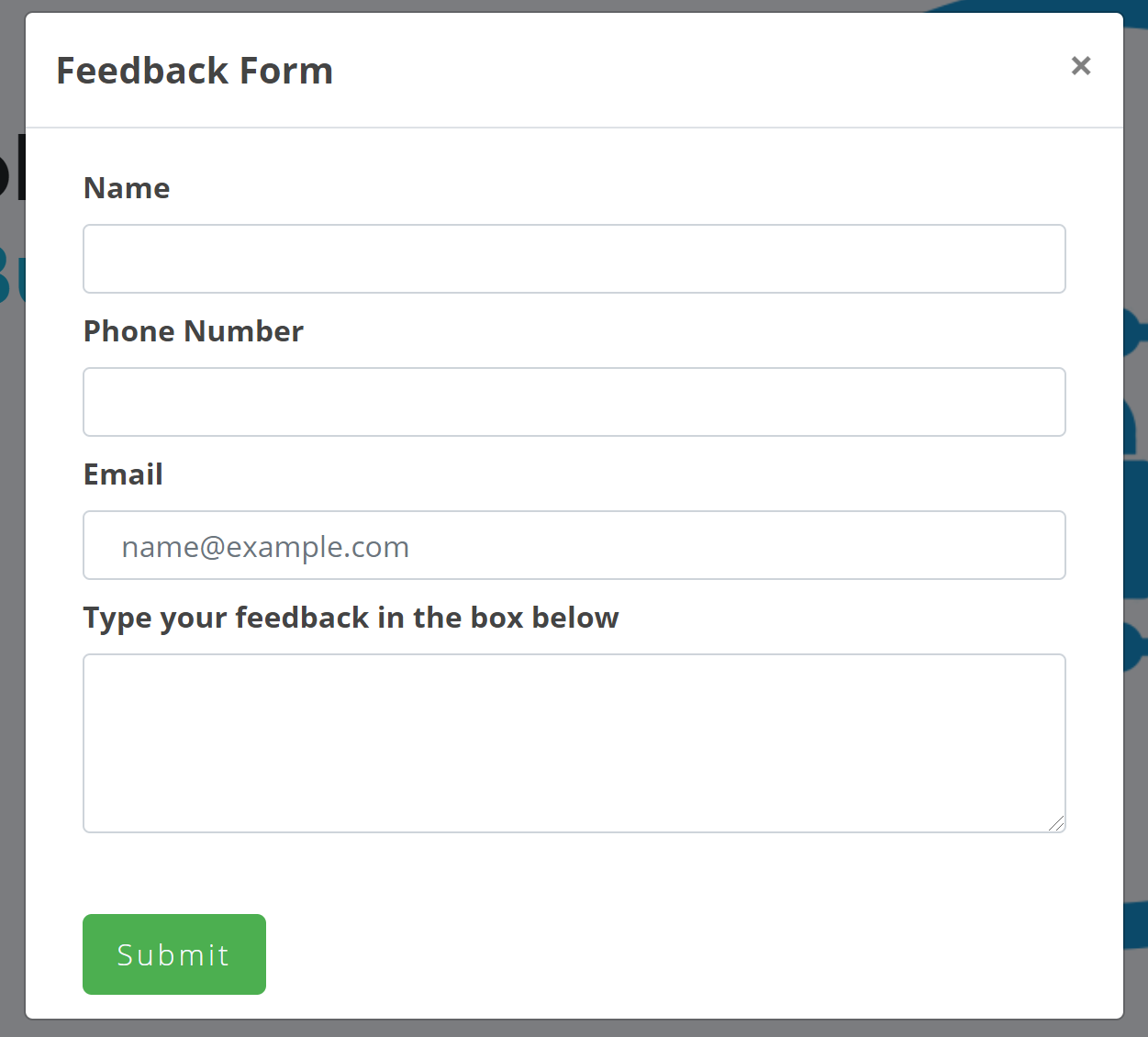
The admin Portal is a consultant-side tool built into the UI, which allows modification, addition and deletion of the questions which are used for the assessment and analysis of the client’s security posture. It also contains options to see MI and para data associated with the data collected across all clients, which can be used to include further improvements to the system.

As the admin clicks this tab it will ask the admin for the credentials to login as shown below:



4. **Feedback:** This tab is for the user to provide any concerns they have or issues they are facing while using the website. They can write their feedback and submit which can be used for later improvement of the website. The feedback can also be a positive comment by the user or their experience.

The below screenshot shows how the feedback option appears on the website. The white space is empty for the user to add his/her comment.



The below screenshot displays the home page of the website hence listing all the tabs described above.



## 3.1 Navigation Flow Diagram for the Web Application

The navigation flow diagram shows the systematic flow of the operations taking place once a tab is clicked by the users (A person accessing the website, Person who is Admin accessing the website). The admin user has the opportunity to manage the mitigation strategies, add a new question or recommendations to the existing ones and view the statistics of the responses given by the users. Whereas the other User can answer the questions of several mitigation strategies to get the maturity level in two ways:

A picture containing parking, meter

Description automatically generated

1. **Get started**: In this way Users can only answer the core questions of all the mitigation strategies. The end result for the user will be the maturity level for all the 8 mitigation strategies along with mitigations or recommendations for all the undesired responses which they can use to protect their system from threats by implementing it on to their computers/systems.

2. **Assess**: By choosing this the user can choose any of the 8 mitigation strategies and answer both core/extended questions and get the maturity level along with recommendations.

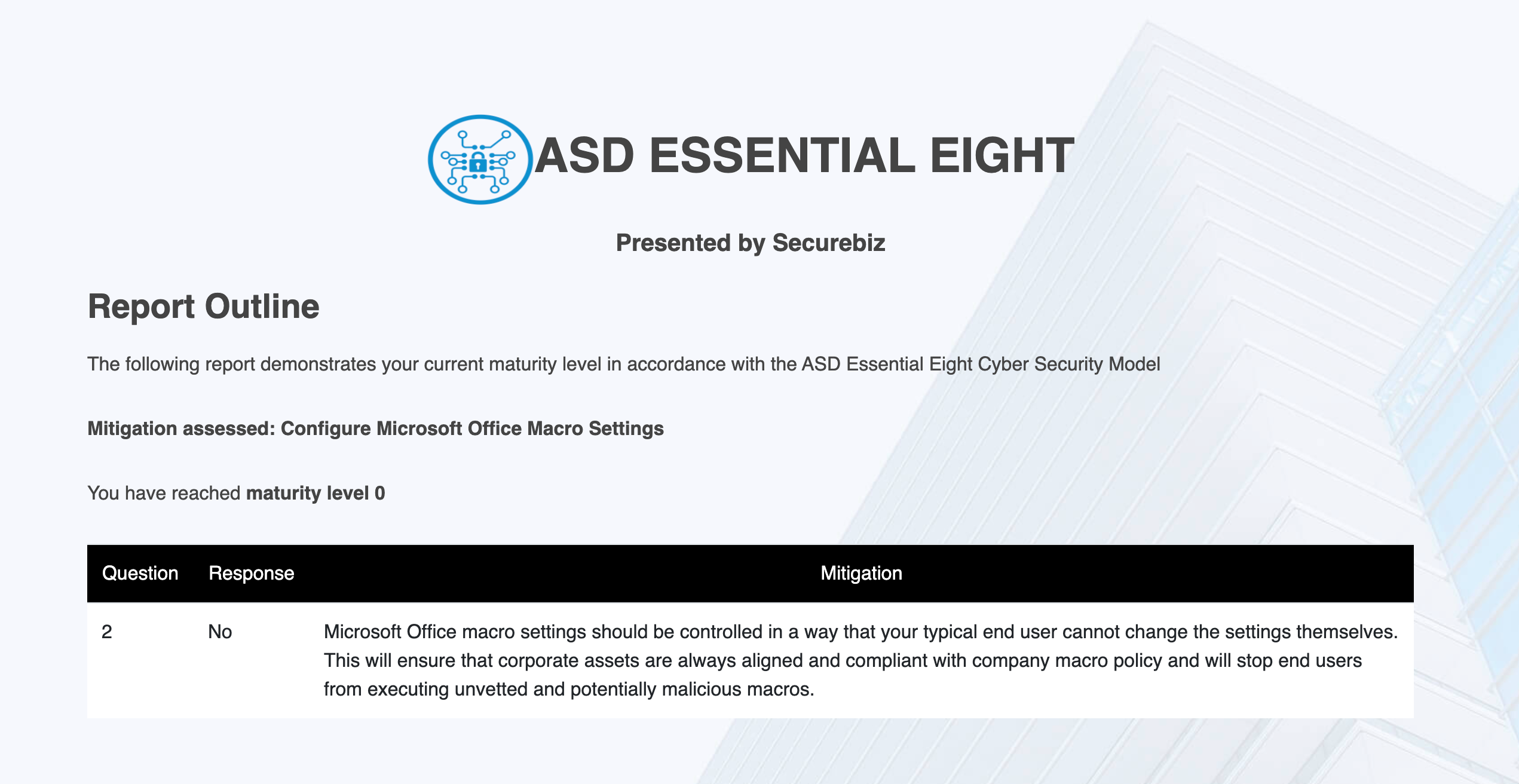
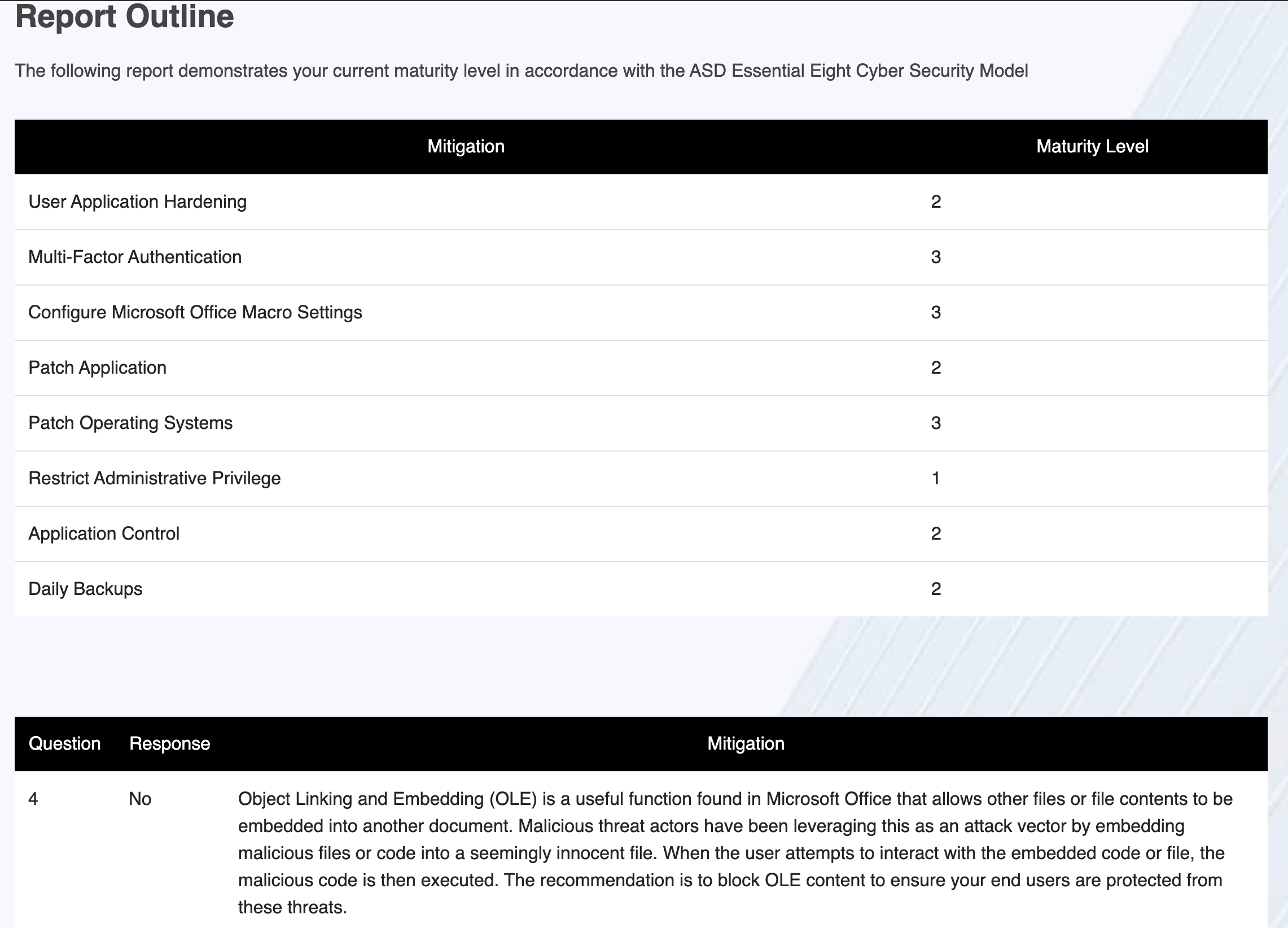
3.**Feedback:** User has option to share their views regarding the website, give recommendations and positive feedback which the admin can implement later in the website.User clicks on feedback option, add the comment and submit which is further stored into the database.

4. **Admin Portal:** This portal can only be accessed by the admin, the admin first required to login to access this option. It provides two option to the admin, First manage mitigation strategy and second Statistics. The manage mitigations strategy provides admin with three options: Add Mitigation, Edit Mitigation and List Mitigation. By this the admin selects any mitigation strategy out of three and can choose to insert new question, along with description and mitigation for the undesired answer. Statistics option provides the Admin to analyse the stats of the website based upon the selection of the user and user preferences.

## 3.2 Presentation-Report Page and PDF

Majority of the statistics are implemented, currently an overall design of the report page has been developed. This design varies depending upon whether the user choose to select a particular mitigation strategy for answering the question of select Gets started button to answer all the core questions.

The screenshots below show various formats in which Report Page or PDF is presented to the user.

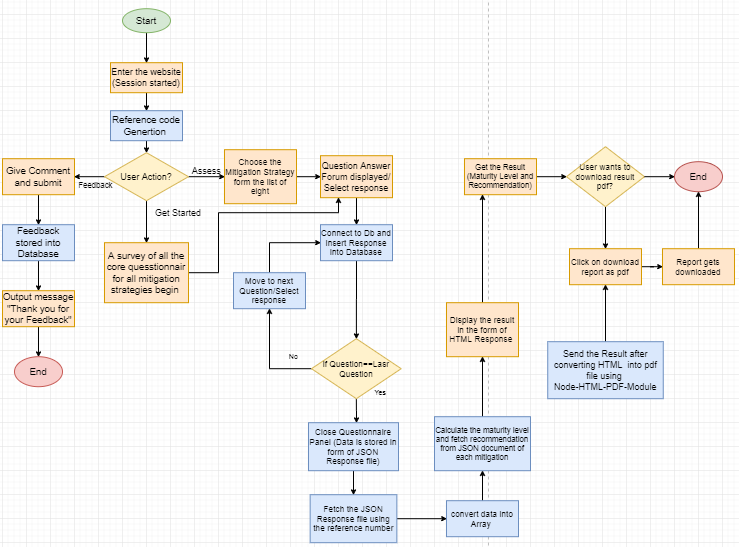
 

There are two types of formats of the report page. First one is for the individual Mitigations strategies and the second style is for the get started button. There are other improvements planned to be integrated in the report page and PDF in order to improve its presentation, such as graphical representation which is yet to be implemented.

To have a detailed understanding of the report page and PDF formation please have a look at the document attached below (Named: Report and Pdf)

## 

## 4.Project process and Process diagram



The process diagram explains the navigation flow of how the process is functioning in the back-end and front-end when a particular request is generated by the user.

When a Assess is clicked it asks for choosing the mitigation strategies from the list of given 8 options then the questions are displayed by making a connection with MongoDB database using a Get API from the database which continuously check for the last question using the code built in the back-end. If question is last the question panel is closed and the then to display the result the response JSON file is fetched from the backend and data is fetched in put into Arrays and maturity level calculations are done and hence the report is generated and sent to the user in the form of HTML and Pdf which user can view and download(yet to be implemented).

When feedback is clicked user is asked to write their comment and submit which will be further stored into database and can be further used by the admin to improve the website.

When user clicks on Get Started, A survey begins where they answer all the core questions for all eight mitigation strategies and then a report is generated at the end.

## 5.Technologies Used

**ReactJS:** ReactJS is a Javascript framework that can be used to develop a single page application (SPA), SPA offers a much more responsive interaction with web users than HTML and Javascript can provide. In development, codes can written in Javascript and JSX (mix of JS and HTML) and compiled to a Javascript bundle that is served into a HTML page. Interaction of the SPA comes from the JS bundle file.

**SASS/CSS:** CSS (Cascading style sheet) is a style sheet language used to describe presentation of the content written in HTML. CSS has been integrated into HTML to enhance the readability and interest of the client. SASS is superset of CSS that is used in the ReactJS code base, on compile SASS would transform into CSS.

**Bootstrap:** Bootstrap framework is used to create a responsive front end for mobile first front-end web development. It includes built in design templates for CSS and HTML including forms, buttons etc. Bootstrap is used in the project to ease and speed up front end development.

**JavaScript:** JavaScript is a client scripting language used for processing and creating dynamic content on web page. It is used for creating functions to perform dynamic functions upon user request. All validations on front end have been achieved though JavaScript.

**NodeJS:** NodeJS is an open source JavaScript framework that is used for building network applications. It has built in packages and functions to handle incoming request and send responses. Node.js has advantages over other backend languages such as ASP, PHP etc. Backend languages such as ASP can handle only request at a time. It only accepts second request after first one has been processed and response sent back to Client whereas Node.Jjs can handle and process multiple requests at a time. Hence, Node.js is used since it’s lightweight and perfect for data intensive real time applications.

**Database**

MongoDB (Atlas): All the data used in this application needs to be stored and retrieve in form of key: value pair. Hence NoSQL language solves the above problem. MongoDB is used to store and retrieve the data in documents. Data is stored in JSON format. Moreover, MongoDB drivers can be loaded in NodeJS. To work MongoDB, Mongo DB Compass has been used which act as a Client for MongoDB with built-in GUI integration.

## 6. Database Design

## 7. Application Security

## 8. Software Requirements