



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

‘Web Development Internship - Publicis Sapient’

Submitted by

V SAISRI (PES1201701763)

January - April 2021

Carried under the guidance of

Swati Sehrawat

Senior Associate Agile Program Management
Core Teams, Talent

Publicis Sapient

Bengaluru



CERTIFICATE

This is to certify that the Report entitled

‘Web Development Internship - Publicis Sapient’

is a bonafide work carried out by

V SAISRI (PES1201701763)

In partial fulfillment for the completion of 8th semester internship work in the Program of Study B.Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan – Apr. 2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 8th semester academic requirements in respect of Internship work.

*Signature with date & Seal
(Dr/ Prof)
Internship co-ordinator*

*Signature with date & Seal
Dr. Anuradha M
Chairperson*

Name and signature of the examiners:

- 1.
- 2.



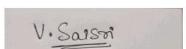
DECLARATION

I, V Saisri, hereby declare that the report entitled, '**Web Development Internship - Publicis Sapient**', is an original work done under the guidance of **Swati Sehrawat**, Senior Associate Agile Program Management, Core Teams, Talent, **Publicis sapient** and is being submitted in partial fulfillment of the requirements for completion of 8th Semester course work in the Program of Study, B.Tech in Electronics and Communication Engineering.

PLACE:Bangalore

DATE: 24/5/2021

NAME AND SIGNATURE OF THE CANDIDATE

1. V Saisri - 



ABSTRACT

It's impossible to imagine a world without the internet. And the proliferation of gadgets. The internet is unquestionably expanding, with an increasing number of devices connecting to the internet on a regular basis and new types of devices emerging.

The Internet has effectively created a global marketplace where people from all over the world can buy and sell a wide range of services and products. We've arrived at a stage where we can connect with the internet in new and enhanced ways thanks to the advent of web applications and the Internet of Things.

Previously, only intranet and internet web pages and websites were concerned with online growth. Nowadays, it's all about creating online applications that are more complex and scaled than conventional smartphone and desktop apps. It can be said that designing complex business applications for both companies and customers is an artistic method of creating online applications.

The internet of today is a never-ending universe in and of itself. Photos, videos, and other multimedia materials, as well as web applications and web sites, are all included. What the user doesn't realise is how browsers and web technologies communicate, and this is what makes all the difference.

Web technologies have progressed over time, allowing web designers and developers to create entirely new levels of immersive and useful web experiences for users. The web as we know it today is the product of the ongoing efforts of a diverse community of web designers who contribute to the advancement of cutting-edge technology.

CSS3 , HTML 5, JavaScript, React JS, AngularJS and different technologies are presently being utilized in internet creation. These structures additionally assure that the internet site or cellular app is well matched with all fundamental net browsers.

The boom of programming software programs and era is accelerating. Different varieties of improvement had been used, inclusive of each frontend and backend improvement.

The mission of the following report is to not only throw light on my learning at Publicis Sapient but to simultaneously show appreciation towards the art of Web Development.



ACKNOWLEDGEMENTS

Firstly, I would really like to express great gratitude toward my instructor throughout the Internship period, Prashanth Puranik. Prashanth sir has now no longer simply been a tremendous instructor in phrases of being capable of delivering his information to us however has been an inexplicable mentor all through the schooling period. His patience, perseverance and determination he has proven has clearly stimulated me to grow to be greater ambitious and work harder.

Secondly, a huge thank you to the professional colleagues at Publicis Sapient. *Sangeeta Debnath* has been a wonderful POC and ensured that I always felt like I belonged at Sapient . Her efforts to bring the whole team together despite the work from home situation is truly commendable. Our mentors *Manish Shekhwat , Manish Dhar, Shobhit Sarin, Vishav Rattan, Ipsita Basak, Amit Chawla, Naman Jain and Kota Sai Siddharth* were always available to help us on the journey to becoming good front end developers. A special note of thanks to all of them because their inputs and insights have been a huge help in developing an understanding of the subject and the practical aspects associated with it.

Publicis Sapient, thank you for giving me the opportunity to intern at your company and welcome us into the corporate world in an extremely friendly and supportive manner.

Lastly, I would like to take this opportunity to thank *Dr. Anuradha M*, Chairperson, Department of Electronics and Communication Engineering, PES University, for all the knowledge and support I have received from the department. We would also like to thank *Dr. B.K. Keshavan*, Dean of Faculty, PES University for his help.

We are deeply grateful to *Dr. M. R. Doreswamy*, Chancellor - PES University, *Prof. Jawahar Doreswamy*, Pro-Chancellor - PES University, *Dr. Suryaprasad J*, Vice-Chancellor - PES University for providing us with various opportunities and enlightenment every step of the way.



TABLE OF CONTENTS

1. INTRODUCTION	8
1.1 Tech Stack	8
1.1.1 VS Code	
1.1.2 Apache server	
1.1.3 Bitbucket	
1.1.4 JIRA	
1.1.5 Lighthouse	
1.1.6 XAMPP	
1.1.7 W3C Validator	
1.1.8 NVDA	
1.1.9 ESLint	
1.1.10 Sass	
1.1.11 Jest	
1.1.12 Postman Tool	
1.2 About the company	12
2. LITERATURE SURVEY	13
2.1 Bootstrap	13
2.2 MDN	14
3. PROJECTS UNDERTAKEN	15
3.1 The Meetings Application	15
3.1.1 Introduction	
3.1.2 Working	
3.2 Sabka Bazaar App	27
3.2.1 Introduction	
3.2.2 Working	
3.3 The Audit App	32
3.3.1 Introduction	
3.3.2 Working	
3.3.3 Tools	
3.3.4 Concepts	
4. CONCLUSION	37
5. REFERENCES	37



LIST OF TABLES AND FIGURE

Figure 1 VS Code	8
Figure 2 Apache Server	8
Figure 3 Bitbucket	8
Figure 4 JIRA	8
Figure 5 Lighthouse	8
Figure 6 XAMPP	8
Figure 7 W3C Validator	8
Figure 8 NVDA	8
Figure 9 ESLint	8
Figure 10 Sass	8
Figure 11 Jest	8
Figure 12 Postman	8
Figure 13 Register as a User screen on the Meetings Application	16
Figure 14 Register as a User screen on the Meetings Application (Mobile View)	16
Figure 15 Login as a User screen on the Meetings Application	17
Figure 16 Login as a User screen on the Meetings Application (Mobile View)	17
Figure 17 Calendar View of the Meetings Application (Desktop)	19
Figure 18 Calendar View of the Meetings Application (Mobile)	19
Figure 19 Filter and Search Meetings View of the Application (Desktop)	20
Figure 20 Filter and Search Meetings View of the Application (Mobile)	21
Figure 21 The Add a meeting page of the Application	22
Figure 22 The Add a meeting page of the Application (Mobile)	23
Figure 23 Teams Page (Desktop)	24
Figure 24 Teams Page (Mobile)	25
Figure 25 Register as a User screen on the Sabka Bazaar Application	28
Figure 26 Login as a User screen on the Sabka Bazaar Application	29
Figure 27 Home Page of the Sabka Bazaar Application	29
Figure 28 Products Page of the Sabka Bazaar Application	30
Figure 29 Empty cart view of the Sabka Bazaar Application	31
Figure 30 Cart view of the Sabka Bazaar Application	31
Figure 31 Atomic Design Pattern	32
Figure 32 Login Page of Audit App	33
Figure 33 Sign Up Page of Audit App	34
Figure 34 Start Audit Page of Audit App	34
Figure 35 Search Listing of Audit App	35
Figure 36 Search Landing Page of Audit App	35
Figure 37 Questionnaires Page of Audit App	36



1. INTRODUCTION

1.1 Tech Stack

The choice of Technology stack or the components within the Tech stack , depend on system requirements such as speed , throughput and scalability. The tech stack hints at the overall strength of the application. The programming languages, frameworks, and resources that a developer will need to interact with the application are summarized in a tech stack.



Fig.1. VS Code



Fig.2. Apache Server



Fig.3. Bitbucket



Fig.4. JIRA



Fig.5. Lighthouse



Fig.6. XAMPP



Fig.7. W3C validator



Fig.8. NVDA



Fig.9. ES Lint



Fig.10. Sass



Fig.11. Jest



POSTMAN

Fig.12. Postman



1.1.1. Visual Studio Code

Microsoft's Visual Studio Code(Fig.1.) is a freeware source-code editor for Windows, Linux, and macOS. Debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git are among the features. Users can customize the theme, keyboard shortcuts, and preferences, as well as install plugins that add new features. The platform has thus been used to execute all the applications due to its high range of features.

1.1.2 Apache Server

The Apache HTTP Server(Fig. 2.) is a free and open-source web server that uses the internet to distribute web content. It's known as Apache, and it quickly became the most successful HTTP client on the internet after its release. While it is commonly assumed that Apache's name comes from its development history and process of change through patches and modules, this was proven incorrect in 2000. It was revealed that the name originated from the respect of the Native American tribe for its resiliency and durability. Apache Server has hence been used to run our applications on the browser using http.

1.1.3 Bitbucket

For technical teams, a Git repository management solution such as Bitbucket(Fig. 3.) is usually used.. It provides a central location for managing git repositories, collaborating on source code, and navigating the development process. It has some fantastic features, such as:

- Access control, which allows you to limit who has access to your source code.
- Workflow control to enforce a project or team workflow.
- Pull requests with in-line commenting for collaboration on code review.
- Jira integration for full development traceability.

1.1.4 JIRA

Jira(Fig. 4.) has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.

Jira integrates with a variety of add-ons so the QA's testing slides seamlessly into the software development cycle. Teams can test effectively and iteratively. QA teams use Jira issues, customized screens, fields, and workflows to manage manual and automated tests.



1.1.5 Lighthouse

Lighthouse(Fig. 5.) is an open-source, automated web page quality-improvement tool. You can use it on any web page, whether it's public or requires authentication. Efficiency, accessibility, progressive web apps, SEO, and more are all audited.Lighthouse is available as a Chrome DevTools plugin, a command-line tool, and a Node module. Lighthouse receives a URL to audit, performs a series of audits on the website, and then produces a report on how well the page performed. Then, as an example of how to change the page, use the failing audits. Each audit has a reference document that explains why the audit is relevant and how to correct it.

1.1.6 XAMPP

XAMPP (Fig. 6.) stands for cross-platform, Apache, MySQL, PHP, and Perl, and it helps you to build a WordPress site on your computer's local web server. The “cross-platform” component refers to the fact that this simple and lightweight approach works on Windows, Linux, and Mac. Since WordPress isn't a stand-alone program, XAMPP includes two key components for its setup: Apache and MySQL.

1.1.7 W3C Validator

The W3C's Markup Validator(Fig. 7.) is a free tool for checking the authenticity of Web documents. Markup languages like HTML and XHTML are used to create the majority of Web documents. Technical specifications, which typically involve a machine-readable formal grammar, describe these languages (and vocabulary). Validation is the process of reviewing a document against these restrictions, and this is what the Markup Validator does.Validating Web documents is a crucial move that can significantly improve and ensure their accuracy while also saving time and money (read more on why validating matters).

1.1.8 NVDA

NVDA(Fig. 8.) is a program that helps blind and vision-impaired individuals to communicate with the Windows operating system and a variety of third-party applications.

The following are some of the highlights:

Common applications such as web browsers, email clients, internet chat services, and office suites are all supported.

- Over 80 languages are supported by the built-in speech synthesizer.
- Textual formatting, such as font name and size, type, and spelling errors, should be reported when available.



- Text under the mouse is automatically announced, and the mouse location can be indicated audibly if desired.
- Many refreshable braille displays are supported, including the ability to automatically detect many of them, as well as braille feedback on braille displays using a braille keyboard.

1.1.9 ESLint

ESLint(Fig. 9.) is a tool for detecting and reporting trends in ECMAScript/JavaScript code, with the aim of improving consistency and preventing bugs. With a few variations, it is similar to JSLint and JSHint in several ways:

- Esprey is used by ESLint to parse JavaScript.
- ESLint evaluates code patterns using an AST.
- ESLint is totally pluggable; every rule is a plugin, and you can add new rules at any time.

1.1.10 Sass

Sass(Fig. 10) is a CSS preprocessor that adds special features to standard CSS including variables, nested rules, and mixins (also known as syntactic sugar). The aim is to make the coding process easier and faster.A CSS preprocessor is a scripting language that allows programmers to write code in one language and then compile it into CSS. Less and Stylus are two well-known examples of preprocessors. Sass is probably the most common right now.

1.1.11 Jest

Jest is a JavaScript code testing library.(Fig. 11)

It's a Facebook-maintained open source project that's particularly well-suited for testing React code, but it's not limited to that: it can test any JavaScript code. It has the following advantages such as It's fast, it can run snapshot tests, it's opinionated, and it comes with everything you need right out of the box without requiring you to make any decisions.

1.1.12 Postman

Postman(Fig. 12) is a common API client that makes creating, sharing, testing, and documenting APIs simple for developers. Users can build and save easy and complex HTTP/s requests, as well as read their replies, to accomplish this. As a consequence, work is more productive and less boring.When it comes to executing APIs, Postman is extremely useful. You can use them over and over without having to recall the exact endpoint, headers, API keys, and so on once you've entered and saved them.



1.2 About the company

Publicis Sapient is an American digital consulting company, founded in Cambridge, Massachusetts in 1990 by Jerry Greenberg and J Stuart Moore. The company helps to unlock value through a start-up mindset and modern methods, fusing strategy, consulting and customer experience with agile engineering and problem-solving creativity. The experience spanning technology, data sciences, consulting and customer obsession – combined with our culture of curiosity and relentlessness – enables Publicis to accelerate our clients' businesses through designing the products and services their customers truly value. They have spent nearly three decades utilizing the disruptive power of technology and ingenuity to help digitally enable clients' business in their pursuit of next.

Their approach to transformation is grounded in a view of both the company and the customers simultaneously, and a unique fusing of strategy and consulting, experience and engineering with an enduring culture of problem-solving creativity.

In the space between next and now is how. Publicis Sapient strongly believes that how you seize that space is everything.

In conclusion , the last couple of months interning with Publicis Sapient has been nothing short of enlightening. The principles , work ethic of the community and the friendliest colleagues at work have ensured that I have a great experience even from the comfort of home.



2. Literature Survey

2.1 Bootstrap

Bootstrap is a front-end framework and open source project that was developed by a designer and a developer at Twitter. It has since grown to become one of the most common front-end frameworks and open source projects in the world.

Bootstrap was founded by @mdo and @fat at Twitter in mid-2010. Bootstrap was formerly known as Twitter Blueprint before becoming an open-source platform. Twitter hosted the first Hack Week a few months into production, and the project exploded as developers of all ability levels jumped in without any external guidance. For over a year before its public release, it served as the style guide for internal tool creation at the company, and it still does today.

It is a combination of CSS and JavaScript-based design templates for:

- Typography
- Forms
- Buttons
- Navigation
- Other interface components

Bootstrap has had over twenty releases since its initial release on Friday, August 19, 2011, including two big rewrites with v2 and v3. It has introduced responsive features to the entire system as an optional stylesheet with Bootstrap 2. Building on that, they also rewrote the library again with Bootstrap 3 to make it sensitive by default and mobile first. It was rewritten again with Bootstrap 4 to account for two major design changes: a migration to Sass and the adoption of CSS flexbox. Bootstrap is more than capable of building a responsive and mobile-first website, but it isn't the best. W3.CSS, a smaller, quicker, and easier-to-use alternative to Bootstrap, is available.

Following are the main advantages of Bootstrap :

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
- It facilitates users to develop a responsive website.
- It is compatible on most browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.



2.2 MDN

MDN Web Docs (formerly known as MDN) is a free resource for in-depth documentation on web standards such as HTML5, CSS, JavaScript, and more. A community of developers and other contributors keep the MDN Web Docs up-to-date. The entire site is a wiki, meaning anyone can add to or edit the growing collection of documentation and tutorials.

MDN provides useful information for Web technologies, and encourages learning, sharing, and teaching in the open Web community. On MDN, you come together and make things for yourself and for others.

MDN is also a place for Mozilla engineers, such as Gecko or Firefox hackers, add-on developers, and Firefox OS contributors. In the early days there was *DevEdge*, the developer documentation from Netscape which formed the basis of some of MDN's documentation. On October 12, 2004, this popular developer website was shut down by AOL, Netscape's parent company.

Only a few months later, in February 2005, Mitchell Baker was able to rescue DevEdge and reached an agreement with AOL that allowed Mozilla to post, modify, and create new documents based on the former Netscape DevEdge materials. In other words, what happened to the Mozilla source in 1998 finally happened for Netscape's developer documentation as well: It became open source.

- **The Fetch API** provides an interface for fetching resources (including across the network). It will seem familiar to anyone who has used XHR but the new API provides a more powerful and flexible feature set. Fetch provides a generic definition of Request and Response objects (and other things involved with network requests). This will allow them to be used wherever they are needed in the future, whether it's for service workers, Cache API, and other similar things that handle or modify requests and responses, or any kind of use case that might require you to generate your responses programmatically.
- **XMLHttpRequest (XHR)** is a JavaScript API to create AJAX requests. Its methods provide the ability to send network requests between the browser and a server.

The XMLHttpRequest object is a developers dream, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background



3.1 The Meetings Application

3.1.1 Introduction

The Meetings Application allows a user to manage your meetings.

It has additional features such as:

1. Filter to view meetings (past,present or future).
2. Search for meetings based on meeting description.
3. Allows an user to Add a meeting and be part of it.
4. Lets you view your meetings in a calendar view.
5. Can create teams consisting of other users and view all teams you are a part of.
6. A user can also excuse themselves from a team

Certain assumptions were made during the execution of the project such as , a Meeting starts and ends on the same day and does not overlap with any other meetings the user is part of.

A meeting has the following details

- date of meeting
- start and end times (hours 0 – 23, and minutes 0 – 59)
- description (text)
- email ids of attendees (users who will be part of the meeting) / a team's short name
(comma-separated)

Approach followed before execution of the above mentioned Project :

1. Before coding , I made sure to understand the requirements of the application clearly to ensure I was clear about project timelines and working of the application.
2. Coming up with a high level design before developing the application.
3. Decide what components I will have in the application and how they will interact with each other.



3.1.2 Working

a) Register and Login as User in the Meetings App

Calendar Meetings Teams Login

Register

Register for Meets !!

Name

Email ID

Password

Confirm Password

Register Now

Fig.13. Register as a User screen on the Meetings Application

Meetings **≡**

Register

Register for Meets !!

Name

Email ID

Password

Confirm Password

Register Now

Fig.14. Register as a User screen on the Meetings Application (Mobile View)



Fig.15 Login as a User screen on the Meetings Application

Fig.16 Login as a User screen on the Meetings Application (Mobile View)

Figure 13 and 14 depict the UI of both the register screen. of the Meetings Application.
Figure 15 and 16 depict the UI of both the login screen of the Meetings Application.

The required fields are present on both the pages as inputs and necessary validations are also implemented to ensure a safe account for the user.

Once a new user registers on the app , he/she is directly redirected to the login page to enable them to login officially and start using the application.

Additionally for the navigation bar displayed on the top , on devices of width less than 640px, the menu appears collapsed. When the user clicks on the _Hamburger menu icon_ the menu expands to show the list of links and the active link is highlighted on every page. For example, on the Calendar page, the Calendar link in the menu is highlighted.



Achieved the expected working requirement:

Given the static file server running on the port number 80, when invoked with the URL <http://<ipaddress>/register>

- The page loads on a screen of dimensions 1280 x 1024, and also separately on a mobile display of dimensions 375 x 812, and the application shows the user registration page and the below fields accepted in a form with a submit button
 - Email
 - Password
 - Confirm Password (for the register page)
- The above looks similar and works fine when viewed in both Chrome and Firefox
- The design of the page is as per the theme of all other pages whose mock-ups are provided.

The following fields are validated as per rules below

1. Email : valid format for an email
2. Password :minimum 8 characters with at least 1 uppercase, 1 lowercase, 1 digit, and 1 special character.
3. Confirm Password :Exact same value as password input (must match it character-by-character).
4. When the user does not fill in any of the fields, a clear error message appears below the input when the user moves out from the input (blur).
5. When the user does not fill in any of the fields, a clear error message appears below the inputs when the user tries to submit the form - the form is not submitted and focus moves to the first input which has an error.
6. If the password and confirm password fields have different values, an error is reported.

If all inputs are fine the page makes the appropriate request to <http://<backend>:8081/register> along with the registration data. On successful registration, it shows the login page with an alert message on top informing users of successful registration. On error, it shows the registration page with an alert message on top informing the user of error. These messages should automatically close after 10 seconds, or when the user dismisses them.

The page must be accessible - usable using both screen reader (NVDA/VoiceOver) and keyboard. Focus must move from one input to another in source order. On success/error, the alert must gain focus and be read out automatically by a screen reader.



b) The Calendar Page

Fig.17. Calendar View of the Meetings Application (Desktop)

Calendar

21 May 2021

Friday

* You have no meetings scheduled for today

00
01
02
03
04
05
06

Fig.18. Calendar View of the Meetings Application (Mobile)

Meetings

≡

Calendar

21 May 2021

Friday

* You have no meetings scheduled for today

00
01
02
03
04
05
06
07
08



Figure 17 demonstrates the Calendar View of the meetings application. This shows the meetings for a day for the logged in user. By default the meetings for the current day are shown. The day can be selected using a date picker.

As seen , in the navbar , the three tabs on the left are Calendar , Meetings and Teams. Once logged in , the username is also displayed in the navbar.

An option to logout is also provided. When the user clicks on the logout link, the JWT token stored in local storage is removed, and the user is redirected to the login page. An alert appears informing users of successful logout. This should automatically close after 10 seconds, or when the user dismisses it. If user tries to navigate again to any of the authenticated pages, they do not load data (for example, calendar page does not load calendar events)

Achieved the expected working requirement :

Given the static file server running on the port number 80, when invoked with the URL <http://<ipaddress>/calendar>

- The page loads on a screen of dimensions 1280 x 1024, and also separately on a mobile display of dimensions 375 x 812, (Figure 18) and the application shows the calendar page with a list of meetings for the day.
- The user is able to use the date picker to change the date
- The above looks similar and works fine when viewed in both Chrome and Firefox
- When printed, the calendar neatly lists in plain text the meetings for the day

c) Filter and Search Meetings

Figure 19 and Figure 20 show the Filter and Search page in both the desktop and mobile view respectively.

These are the results that appear once the backend returns the matching results. You can leave a meeting, or add attendees using the dropdown of user email ids.

Fig.19. Filter and Search Meetings View of the Application (Desktop)

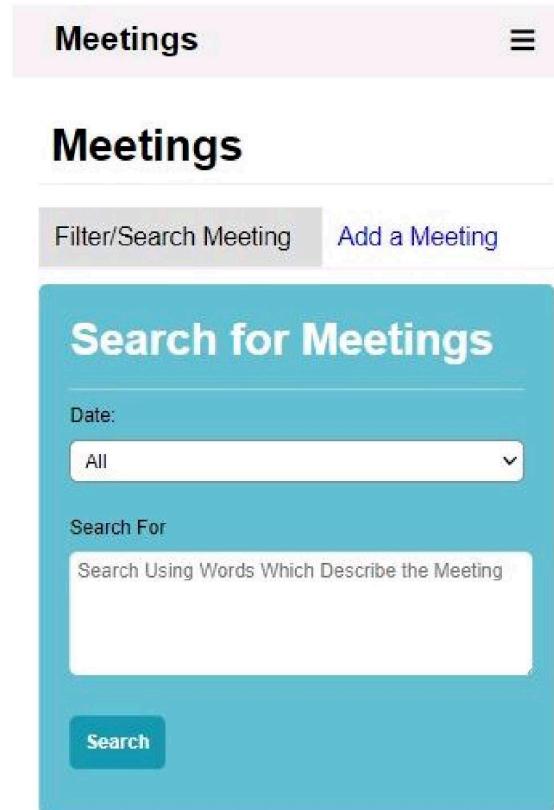


Fig. 20. Filter and Search Meetings View of the Application (Mobile)

- The search / filter dropdown has options "ALL", "PAST", "TODAY", "UPCOMING" (self-explanatory). On selecting an option, the meetings are displayed in chronological order (earliest meeting first). "ALL" is selected in this dropdown by default.
- To match the search terms entered in the textarea , the matched meeting should have AT LEAST ONE of the search terms in its description should show up.

Achieved the expected working requirement :

Given the static file server running on the port number 80, when invoked with the URL <http://<ipaddress>/meetings>

- The page loads on a screen of dimensions 1280 x 1024, and also separately on a mobile display of dimensions 375 x 812, and the application shows the Filter / Search meetings page.
- The user is provided with a form that shall help them search for meetings. The following are part of it
- Date dropdown : this gives the following options
 - Select one (first one)
 - TODAY
 - PAST
 - PRESENT
- The meetings are listed in a list view as desired.
 - The Select member drop down lists registered users as options.
 - The forms inputs have appropriate help description, and designs for errors.



- The above looks similar and works fine when viewed in both Chrome and Firefox.
- The following fields are validated as per rules below
 - Date dropdown - one of TODAY / PAST / PRESENT is selected
 - Search textbox - no validations
- When the above validation does not pass, a clear error message appears below the input when the user moves out from the select input (blur).
- When the user does not select from the dropdown, a clear error message appears below the inputs when the user tries to submit the form - the form is not submitted and focus moves to the select input which has the error.
- If all inputs are fine the page makes the appropriate request to the backend to fetch matching meetings. On success, it shows the list of meetings below the form, with all required details. On error, an alert message on top informing the user of error, and a helpful message to try to rectify this if possible.

d) Add a new Meeting Page

Figure 21 shows the display of the Add a meeting page in the desktop view. In this page , you can add a meeting (i.e. create a new meeting) – when doing so, the email id of participants, or team short name, can be given for attendees (separated by commas).

The screenshot shows the 'Meetings' section of a web application. At the top, there are navigation links for 'Calendar', 'Meetings' (which is currently active and highlighted in grey), and 'Teams'. On the right, there are 'Hello' and 'Logout' links. Below the navigation, the word 'Meetings' is displayed in bold. There are two buttons: 'Filter/Search Meeting' and 'Add a Meeting' (which is currently active and highlighted in grey). The main content area is titled 'Add a New Meeting'. It contains the following fields:

- 'Meeting Name' (text input)
- 'Date:' (date input)
- 'Start Time (hh:mm)' (two dropdown menus for hours and minutes)
- 'End Time (hh:mm)' (two dropdown menus for hours and minutes)
- 'Description' (text area with placeholder 'What is the Agenda of Meeting')
- 'Email ID's of Attendees' (text input with placeholder 'Separate multiple email IDs with a comma')

 At the bottom of the form is a blue 'Add Meeting' button.

Fig.21. The Add a meeting page of the Application



Meetings

Meetings

Filter/Search Meeting Add a Meeting

Add a New Meeting

* All fields are Required

Meeting Name

Date:

Start Time (hh:mm)

End Time (hh:mm)

Description

Email ID's of Attendees

Add Meeting

Fig.22. The Add a meeting page of the Application (Mobile)

Achieved the expected working requirement:

Given the static file server running on the port number 80, when invoked with the URL <http://<ipaddress>/meetings/add>

- The page loads on a screen of dimensions 1280 x 1024, and also separately on a mobile display of dimensions 375 x 812, and the application shows the Add a new meeting page.
- The user is provided with a form that shall help them add a new meeting. The following are part of it
 1. Date (of meeting): Date input
 2. Start time
 - Hours: Dropdown / Range input (0 - 23)
 - Minutes: Dropdown / Range input (0 - 59)
 3. End time
 - Hours: Dropdown / Range input (0 - 23)
 - Minutes: Dropdown / Range input (0 - 59)



4. Description - textbox with 3 lines of input visible
 5. Email ids of attendees - text input
 6. Add meeting button
- The forms inputs have appropriate help description, and designs for errors.
 - All fields are validated as per generally expected behavior.
 - When the above validations do not pass, a clear error message appears below the input when the user moves out from the select input (blur).
 - When the validations do not pass, a clear error message appears below the inputs when the user tries to submit the form - the form is not submitted and focus moves to the first input which has the error.
 - If all inputs are fine the page makes the appropriate request to the backend to add a new meeting. On success, it shows an alert message which does not auto-close. On error, an alert message on top informing the user of error, and a helpful message to try to rectify this if possible. This should not automatically close.
 - The above looks similar and works fine when viewed in both Chrome and Firefox.
 - The page must be accessible - usable using both screen reader (NVDA/VoiceOver) and keyboard.
 - Focus must move from one input to another in source order. On success/error, the alert must gain focus and be read out automatically by a screen reader.

e) The Teams Page

You can view the details of teams you are part of, and add members to them. You can excuse yourself from the team (leave the team), but cannot remove anyone.

A team has a name, a short name (begins with @), a description, and a list of users (identified by their email ids). In order to add a user to a team, a dropdown with user names exists.

Figure 23 shows the Teams page in Desktop View and Figure 24 in mobile view.

The screenshot shows the 'Teams' page with the following details:

- Existing Teams:**
 - Jan XT Batch 3**: @Intern Batch 3, Internship batch 3 training team. Members: aravind@example.com, aravind2@example.com. Includes an 'Excuse Yourself' button.
 - meetind102**: @okkk, meeting now. Members: aravind@example.com, aravind@example.com, aravai2@gmail.com. Includes an 'Excuse Yourself' button.
- New Team Form:**
 - Team Name:** Team Name (input field)
 - Team Short:** Team Short (input field)
 - Team Members:** Team Members (input field)
 - Team Description:** Short Description (input field)
 - Submit:** A blue 'Submit' button.

Figure.23. Teams Page (Desktop)



Meetings ≡

Teams

View and edit teams you are part of

Jan XT Batch 3

@Intern Batch 3

Internship batch 3 training team

Excuse Yourself

Members:
aravind@example.com, aravind2@example.com

Add

meetind102

@okkk

meeting now

Excuse Yourself

Members:
aravind@example.com, aravind@example.com, aravind2@gmail.com

Add

Fig.24. Teams Page (Mobile)

Achieved the expected working requirement:

Milestone 1:

Given the static file server running on the port number 80, when invoked with the URL <http://<ipaddress>/teams>

- The page loads on a screen of dimensions 1280 x 1024, and also separately on a mobile display of dimensions 375 x 812, and the application shows the Teams page.
- The user is shown a list of teams they are part of.
- The Select members is a dropdown with a list of users of the app as options.
- When the '+' button is clicked a form appears in its place that helps use Add a new team
- The forms inputs have appropriate help description, and designs for errors
- The above looks similar and works fine when viewed in both Chrome and Firefox
- The design of the Add a new team form is as per the theme of all other pages.

Milestone 2:

- On page load, a call is made to the backend to fetch the list of teams for logged in users, and on success, the list of teams show up correctly.
- If for some reason the list of teams is not fetched, an appropriate alert message appears with reason for error, and a helpful suggestion to overcome it. This should not automatically close.
- When the '+' button is clicked a form appears in its place that helps users add a new team. Appropriate validations are made on the inputs as the user navigates the inputs.
- When this form is submitted, errors appear again if the form is not valid. If form inputs are fine, the form is submitted to the backend and a new team is created.
- The form and the list of teams must be accessible via - usable using both screen reader



(NVDA/VoiceOver) and keyboard. Focus must move from one team to another - within a team focus must also move to the dropdown and button, before going to the next team in the list.

f) Milestone 3

Milestone 3 deliverables were constant throughout all pages and as follows:

1. Page is structured according to MVC architecture
2. HTML is used for views
3. The Controller is a class that handles user interaction and model updates
4. A Team model is used to store an array of teams, and also store the details of a new team, and handles validation of the properties.
5. Sass is used to style the page
6. A Meeting model stores an array of calendar meetings. A FilterSearchDetails model stores form details and handles validation of the properties and also stores the details of new meetings and handles validation of the properties.
7. ES2015 features are used wherever applicable
 - arrow functions
 - destructuring
 - object literal syntax enhancements
 - classes and modules
 - variables should be declared using const for those not requiring reassignment, and let for those that do.
 - functions should be given sensible defaults for arguments (only wherever applicable)

g) Milestone 4

Milestone 4 deliverables were constants across all the above mentioned pages and were as follows:

1. The page logic must be linted using ESLint or similar linter, and free of lint errors.
2. Jasmine/Jest must be used for unit-testing page logic and code coverage must be at least 80%.
3. Webpack must be used for bundling HTML, SCSS and TS for the page.



3.2. Sabka Bazaar App

3.1.1. Introduction

To help people to have easy access to their grocery shopping..The concept is to make grocery shopping for vegetables and fruits more convenient for customers. This site is for members who do not have enough time to purchase vegetables and fruits. Customers get the freedom to buy organic and hygienic fruits and vegetables at their doorsteps as it becomes time saving for customers.Vegetables and fruits that are exclusive and new are available on the Sabka Bazaar App. Customers would be charged premiums that are lower than those in the industry. Cash on delivery and online payment options are available.

The development of the User Interface and some functionalities is currently our primary focus in this project.

We have used React.js to create the elements and setting up the react application was done using Create React App(CRA).

We have used Bootstrap for styles and Font Awesome for icons and to make Ajax requests we have used the popular axios library.

3.2.2. Working

The Sabka Bazaar App has the following pages:

1. Registration Page

This page accepts the below in a form with a submit button:

- First Name
- Last Name
- Email
- Password
- Confirm Password

The following fields are validated as per rules below

- Email : valid format for an email
- Password : minimum 8 characters with at least 1 uppercase, 1 lowercase, 1 digit, and 1 special character.
- Confirm Password : Exact same value as password input (must match it character-by-character).
- When the user does not fill in any of the fields, a clear error message appears below the input when the user moves out from the input (blur).
- When the user does not fill in any of the fields, a clear error message appears below the inputs when the user tries to submit the form - the form is not submitted and focus moves to the first input which has an error.
- If the password and confirm password fields have different values, an error is reported.



If all inputs are fine the page makes the appropriate request to <http://<backend>:8081/register> along with the registration data.

On successful registration, it shows the login page with an alert message on top informing users of successful registration. On error, it shows the registration page with an alert message on top informing the user of error.

The screenshot shows the 'Signup' page of the Sabka Bazaar application. At the top, there is a navigation bar with the logo 'SABKA BAZAAR' and links for 'Home' and 'Products'. On the right side of the header, there are 'Signin' and 'Register' buttons, and a shopping cart icon indicating '0 items'. The main content area is titled 'Signup' and contains five input fields: 'First Name', 'Last Name', 'Email', 'Password', and 'Confirm Password'. Below these fields is a large red 'Signup' button.

Figure 25: Register as a User screen on the Sabka Bazaar Application

2. Login Page

This page accepts the below in a form with a submit button:

- Email
- Password

The following fields are validated as per rules below

- Email - valid format for an email
- Password - minimum 8 characters with at least 1 uppercase, 1 lowercase, 1 digit, and 1 special character.
- When the user does not fill in any of the fields, a clear error message appears below the input when the user moves out from the input (blur).
- When the user does not fill in any of the fields, a clear error message appears below the inputs when the user tries to submit the form - the form is not submitted and focus moves to the first input which has an error.
- If the password does not meet password requirements, an error is reported saying "Password seems incorrect".

If all inputs are fine the page makes the appropriate request to <http://<backend>:8081/login> along with the login data.



The screenshot shows the login page of the Sabka Bazaar application. At the top, there is a header with the 'SABKA BAZAAR' logo, navigation links for 'Home' and 'Products', and user account links for 'Signin' and 'Register'. A shopping cart icon indicates '0 items'. Below the header, the word 'Login' is prominently displayed in a large, bold font. A sub-instruction 'Get access to your Orders, Wishlist and Recommendations' is present. There are two input fields: 'Email' containing 'test@test.com' and 'Password'. A large red 'Login' button is centered at the bottom.

Figure 26: Login as a User screen on the Sabka Bazaar Application

3. Home Page

This page displays all the categories of items available in the website.

The screenshot shows the homepage of the Sabka Bazaar application. At the top, it features the 'SABKA BAZAAR' logo, 'Home' and 'Products' links, and user account links for 'Signin' and 'Register'. A shopping cart icon shows '0 items'. A large promotional banner for 'INDEPENDENCE DAY SALE' offers 'UP TO ₹70 OFF'. Below the banner, there are several product categories: 'Fruits & Vegetables' (with a shopping basket icon), 'Bakery Cakes and Dairy' (with bread slices), 'Beverages' (with various drink containers), 'Beauty and Hygiene' (with personal care products), and 'Baby Care' (with an image of a baby in a tub). Each category has a descriptive text and a 'Explore [category]' button.

Figure 27: Home Page of the Sabka Bazaar Application



4. Products Page

This page will display the products of the selected category that are available to buy. By clicking on the “Buy Now” button, the item will be added to the cart.



[SignIn](#) [Register](#)

 0 items

- [Home](#)
- [Products](#)

- [Fruits & Vegetables](#)
- [Bakery Cakes and Dairy](#)
- [Beverages](#)
- [Beauty and Hygiene](#)
- [Baby Care](#)

Fresho Kiwi - Green, 3 pcs



3 pc (270g-300g)

Kiwis are oval shaped with a brownish outer skin. The flesh is bright green and juicy with

[Buy Now @ Rs.87](#)

Apple - Washington, Regular, 4 pcs



4 pcs (530g-640g)

The bright red coloured and heart shaped Washington apples are crunchy, juicy and

[Buy Now @ Rs.187](#)

Fresho Pomegranate Peeled, 500 gm



No Preservatives

Pomegranate variety has a glossy, vibrant and soft ruby-red skin. The inside of the fruit

[Buy Now @ Rs.88](#)

Capsicum - Green, 1 kg



Leaving a moderately pungent taste on the tongue, Green capsicums, also known

[Buy Now @ Rs.137](#)

Tomato - Local, Organically Grown, 500 gm



Fresho brings to you an exquisite range of locally grown organic tomatoes, which are

[Buy Now @ Rs.12](#)

Copyright © 2011–2018 Sabka Bazaar Grocery Supplies Pvt Ltd

Figure 28: Products Page of the Sabka Bazaar Application



5. Cart

This page will display all the products in the cart. One can checkout the cart by clicking on “Proceed to checkout”.



My Cart

No items in your cart
Your favourite items are just a click away

Start Shopping

SABKA BAZAAR

Home Products

My Cart (1 item)

	Apple - Washington, Regular, 4 pcs	- + × Rs.187
---	---	--

LOWEST PRICE GUARANTEED You won't find it cheaper anywhere

Promo code can be applied on payment page

Proceed to Checkout Rs.187 >

Figure 29: Empty cart view of the Sabka Bazaar Application

Figure 30: Cart view of the Sabka Bazaar Application



3.3. The Audit App

3.3.1 Introduction

Project Audit is the systematic, often iterative activity of examining and analyzing a project to identify and determine any issues, concerns, challenges and opportunities and their reasons, and to develop recommendations about how to address the problems and keep the project successful.

Currently, project audits are handled through an excel sheet which needs to be updated, circulated and maintained. It is difficult to track the history, and a lot of back and forth happens. The Audit App is a web application that will help to make this process streamlined, flexible and easy to use which keeps a record of the history as well.

Currently, our focus in this project is with respect to creating the User Interface and some functionalities.

We have used React.js to create the elements and have used an Atomic Design Pattern.



Figure 31: Atomic Design Pattern

The file structure is given below:

- Atoms
- Molecules
- Organisms
- Templates

Atoms:

- Basic building blocks of matter, such as a button, input or a form label.
- They're not useful on their own.
- Atoms are the smallest possible components
- They can be applied in any context, globally or within other components and templates

Molecules:

- They are the composition of one or more components of atoms
- Molecules can have their own properties and create functionalities by using atoms, which don't have any function or action by themselves.

Organisms:

- Molecules are combined together to form organisms which make up a distinct section



Templates:

- Different organisms are combined to form a larger section known as template

3.3.2 Working

The Audit App has the following pages:

1. Login/Signup Page

The User signs up if he/she doesn't have an account. The details get added to a Mock API JSON file. Upon logging in, this Mock API JSON file is checked as to whether the User details and password are correct. If it is correct, the User gets logged in.

Respective validations are done for each input entry of the form.

An error message arises in the Login form if:

- a. An input is left empty upon submission
- b. Email does not have an '@' symbol
- c. Password is incorrect

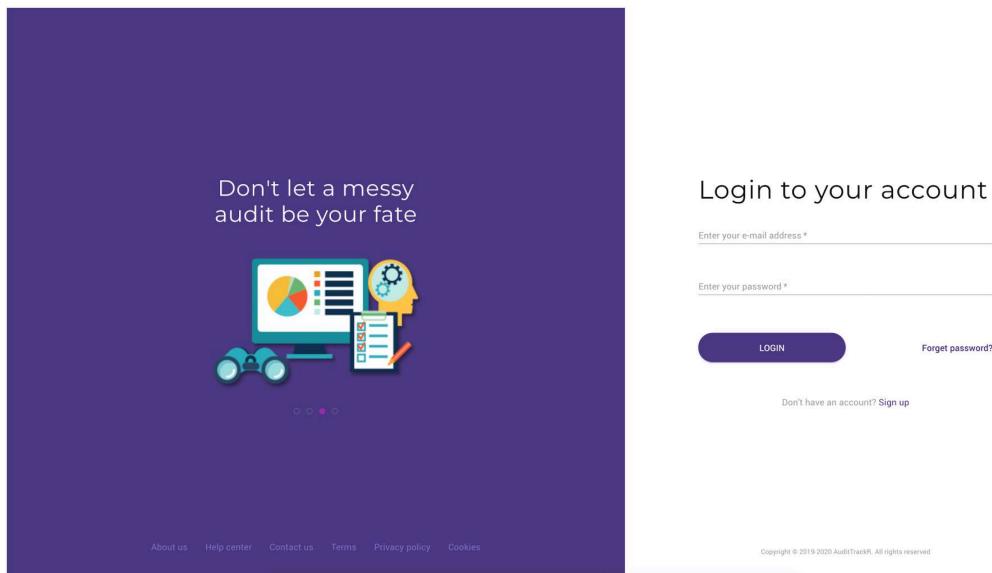


Figure 32: Login Page of Audit App

Similarly for the Sign Up form, an error message arises if:

- a. An input is left empty upon submission
- b. Email does not have an '@' symbol
- c. The 'Confirm Password' does not match with the inputted 'Password'



The sign-up page features a dark purple header with the word "LOGO" in white. Below it is a central message: "Don't let a messy audit be your fate". To the right is a form for creating a new account. It includes fields for "Enter your username", "Enter your e-mail address", "Enter your password", and "Enter your password again". There is also a checkbox for accepting "Terms & Conditions". A "SIGN UP" button is at the bottom left, and a link "I already have an account" is at the bottom right. The footer contains links for "About us", "Help center", "Contact us", "Terms", "Privacy policy", and "Cookies". Copyright information "Copyright © 2018 - 2019 Publicis Sapient. All rights reserved." is at the very bottom.

Figure 33: Sign Up Page of Audit App

2. Start Audit Page

From this page, the User will fill the form in order to start a new audit. Various details of the new project such as the Project Name, the Category of the project and its reviewer must be filled in. For this form respective validations have been done. If any entry is left empty upon submission, an error message arises.

Upon successful creation, The User will be redirected to the Audit Questionnaires page.

The start audit page has a dark sidebar on the left with a "Dashboard" tab highlighted. The main area is titled "START NEW AUDIT" with a sub-instruction: "Start creating an audit to add pace to your business, don't let a messy audit be your fate.". It contains four input fields: "Enter an audit name *", "Enter the project name *", "Select a category", and "Enter reviewer id *". A "START NEW AUDIT" button is at the bottom. The top navigation bar includes a search bar, a user icon, and a "START AUDIT" button.

Figure 34: Start Audit Page of Audit App

3. Search Listing/Landing Page:

The Search Listing is a drop-down. Here, based on our search input, relating search results are retrieved from the Mock API.

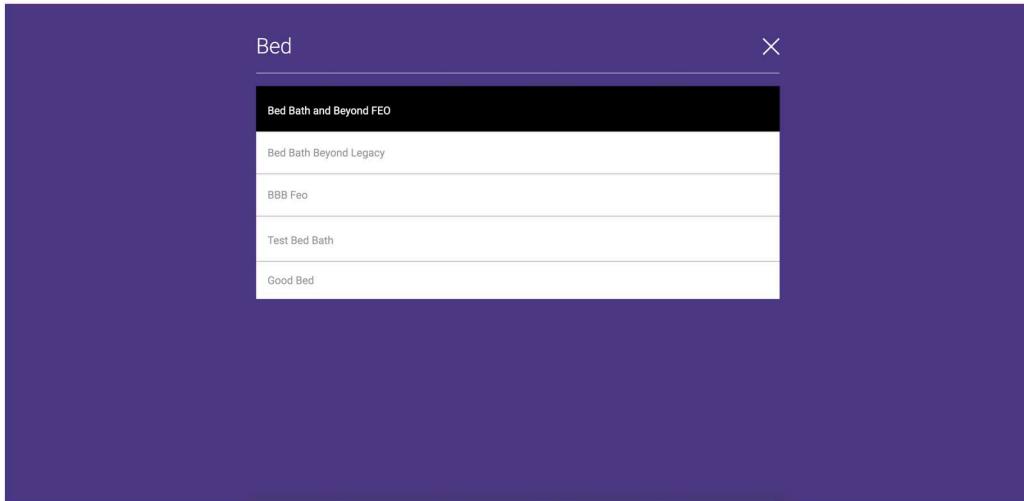


Figure 35: Search Listing of Audit App

Upon searching, the User lands on the Search Landing page where relating results are retrieved and displayed along with their Status, Reviewers and Progress. Each search result be deleted if required. It can be filtered further based on Progress, Status and Category. It can also be ordered in an Ascending or Descending order with respect to the Project Name or Progress.

Figure 36: Search Landing Page of Audit App

4. Questionnaires Page

- User will land on this page after creation of a new audit or if the user needs to edit his previously created audit.
- On this page, based on the predefined questionnaire template, the User will see different categories with associated questions.
- From this page, the User can share the particular audit with other reviewer/collaborator(s).



- User can check the history of the audit.
- User will also see how much percentage of the audit is complete.
- User can choose to close the audit with description.
- User will also see information like what questions have been reviewed by reviewer.

Figure 37: Questionnaires Page of Audit App

3.3.3 Tools

Some of the Tools used for the Audit App are:

1. Theming and Style Components:

Theming is used for code reusability. Styled Components is a flexible way to style React components with CSS. It provides out-of-the-box theming support using a wrapper component called, `<ThemeProvider>`. This component is responsible for providing the theme to all other React components that are wrapped within it. This way multiple elements can be styled using colors and styles defined once.

2. Storybook:

Storybook is a tool for UI development. It makes development faster and easier by isolating components. This allows you to work on one component at a time. You can develop entire UIs without needing to start up a complex dev stack, force certain data into your database, or navigate around your application.

3.3.4 Concepts

Some of the concepts incorporated in the Audit App are:

1. Responsiveness:

All the components, ranging from the atoms to the templates have been made responsive and hence are compatible to different devices. This has been done using units such as 'em', 'rem', '%' and 'vw' and also using Media Queries where required.



2. Accessibility:

All the components have been made keyboard accessible and accessible via screen-readers by using ‘aria’ tags where required.

4. Conclusion

The Meetings Application, Sabka Bazaar App and The Audit App were all completed successfully and fulfilled the given scope. Non-Functional Requirements (NFRs) such as Responsiveness and Accessibility were also included and all the feedback given by our mentors were implemented.

5. References

- [1] Fetch API [online]
Available: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
- [2] About - Bootstrap [online]
Available: <https://v4-alpha.getbootstrap.com/about/history/>
- [3] MDN Web Docs — What It Is and How to Use It , *January 31, 2020* [online]
Available: <https://www.freecodecamp.org/news/the-mozilla-developer-network-what-it-is-and-how-to-use-it/>
- [4] XHR(XMLHttpRequest)[online]
Available: [https://developer.mozilla.org/en-US/docs/Glossary/XHR_\(XMLHttpRequest\)](https://developer.mozilla.org/en-US/docs/Glossary/XHR_(XMLHttpRequest))
- [5] Styled Components [online]
Available: <https://styled-components.com/docs/basics>
- [6] React Documentation [online]
Available: <https://reactjs.org/docs/getting-started.html>
- [7] Sass Documentation [online]
Available: <https://sass-lang.com/documentation>
- [8] Webpack [online]
Available: <https://webpack.js.org/concepts/>
- [9] Accessibility Guide [online]
Available: <https://www.w3.org/TR/wai-aria-1.1/#introduction>
- [10] Theming [online]
Available: <https://css-tricks.com/theming-and-theme-switching-with-react-and-styled-components/>
- [11] Storybook [online]
Available: <https://storybook.js.org/docs/react/get-started/introduction>