

# Internship Documentation

Saurabh Chidrewar

SDE Internship at **Babel Pte Ltd**

Oct 2021 - Feb 2022

---

## Responsibilities performed:

- Built efficient and reusable front-end systems that drive Babel's state-of-the-art web applications
- Collaborated with Product Designers, Marketing Managers, Product Managers, and Software Engineers to deliver compelling user-facing products
- Interact with other team members to incorporate their innovations and vice versa
- Identify and resolve performance bottlenecks and scalability issues in the front end systems

## Objective:

Designing front end systems constituting dashboards and charts that can be plotted using the given data (the data can be fetched from any source) using ReactJs.

## Abstract:

In this project, we've used the power and efficiency of ReactJS Javascript library to design front end systems that can be used to display web designs. We've also plotted different bar charts and line charts using ChartJs (open-source JavaScript library for data visualisation). Different interactive images (gifs) and tables are also incorporated in the project. This project is created referring to the Figma design provided by the project manager.



The **Figma design** referred in this project can be viewed here:

[Mobility Solutions - Tyre Team & Client Journey](#)

The complete code and dependencies are available on the github repository provided by the organisation.



The **GitHub repository** can be found here:

[Mobility Solutions - GitHub](#)

The Daily Standup and progress excel sheet helped in managing the tasks done and also tracking the blockers and getting them resolved.



The Daily Standup Excel Sheet can be viewed here:

[Daily Standup - Saurabh Chidrewar](#)

# INDEX

Sr.	Title
1	<b>Introduction to Web Designing</b>  <input type="checkbox"/> Front End <input type="checkbox"/> Back End <input type="checkbox"/> Databases
2	<b>Technologies used in the Project</b>  <input type="checkbox"/> ReactJS <input type="checkbox"/> ChartJS <input type="checkbox"/> Git and Github <input type="checkbox"/> React Developer Tools <input type="checkbox"/> Pesticide
3	<b>Demonstration</b> of the web design (Video Link)
4	<b>Learning resources</b>
5	<b>Conclusion</b>

## Introduction to Web Designing:

### The Internet:

The Internet is the *global system of interconnected computer networks* that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. Internet was *started as a US Military project* named ARPANET directed by Robert Taylor and managed by Lawrence Roberts.



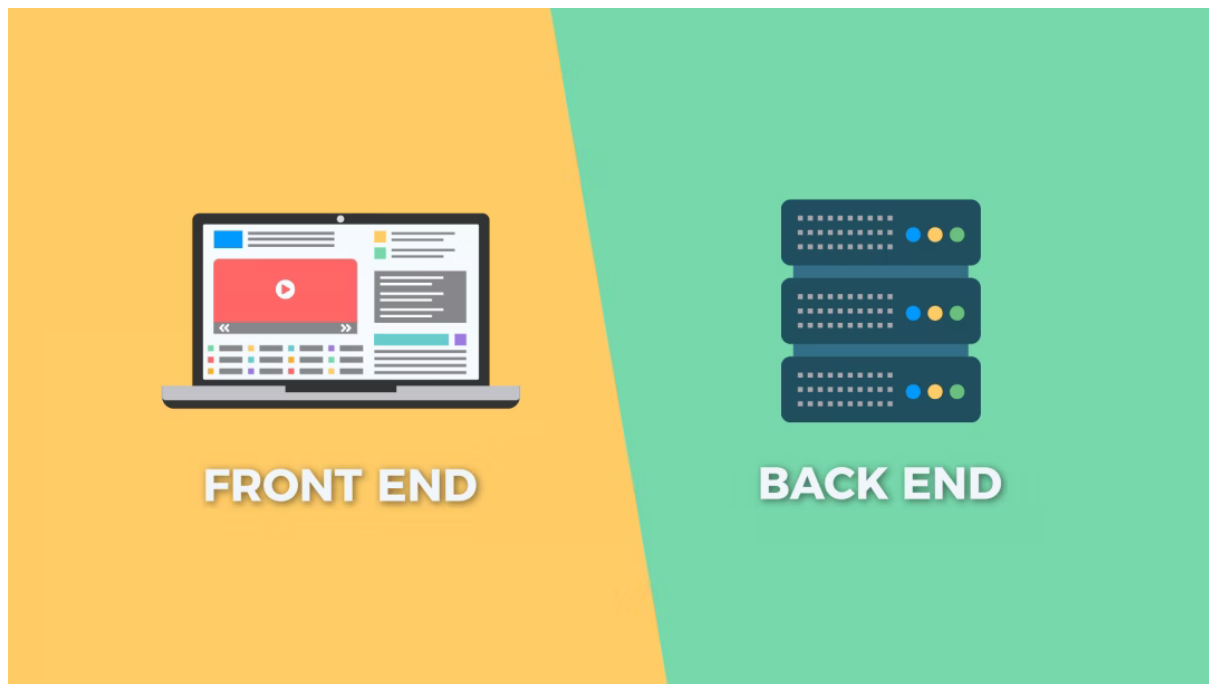
### WWW:

The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.

Now, to use these services, we need some softwares which can build things and hence help our work get done on the web. Web technologies which we will be covering to build those things are HTML, CSS, Javascript etc. The primary focus for the front-end will be on the ReactJS which powers a huge number of devices and websites.

*Web programming*, also known as *web development*, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization.



### Front-end System:

A front-end system is part of an information system that is directly accessed and interacted with by the user to receive or utilise back-end capabilities of the host system. It enables users to access and request the features and services of the underlying information system. The front-end system can be a software application or the combination of hardware, software and network resources.

A front-end system is primarily used to send queries and requests, and receive data from the back-end system or the host information system. It serves or provides users with the ability to interact and use an information system. Typically, front-end systems have very limited computational or business logic processing capabilities and rely on the data and functions from the host system. However, some advanced level front-end systems do maintain copies of data, such as a duplicate of each transaction sent to the back-end system.

A front-end system may include or consist of textual or graphical user interface (GUI) and/or a front-end client application that is connected by the back-end system.

Front-end web development is everything involved in programming the user interface of a web application. Typically it refers to the Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript portion of web site production as opposed to the database or server-side programming. It encompasses everything from building a simple page of HTML text to creating complex, responsive HTML5 websites designed to be accessed via various different browsers, devices and screen sizes.

## Back-end System:

Back-end is the part of the website which deals with the core functioning of the website and is hidden to the user for user's safety. Users shouldn't know what is happening on the website, this is the concern of the back-end developers. Having a back-end makes the website more dynamic. When users interact with the website which involves back-end, it makes the creators easy to involve with users for the main purpose of the website. Back-end involves maintaining the database of various users, helping them to get things done through the various tools and services developed by the programmers of the back-end. Common objectives of the back-end are to involve users with the website, maintaining the proper database for various users.

## Databases:

A database is an organised collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just a database.

Data within the most common types of databases in operation today is typically modelled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organised. Most databases use structured query language (SQL) for writing and querying data.



### Front End

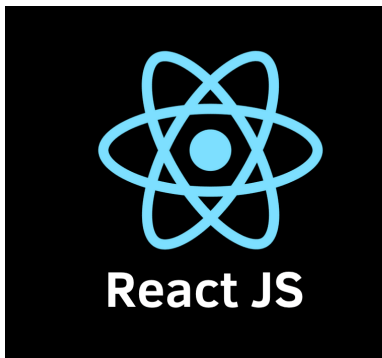
- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation



### Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

## ReactJS :



React.js is the most popular front-end JavaScript library for building Web applications. React.js or Reactjs or simply React are different ways to represent React.js.

Most fortune 500 companies use ReactJS.

React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components. React was first created by Jordan Walke, a software engineer working for Facebook. React first deployed on Facebook's newsfeed in 2011 and on Instagram.com in 2012.

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application. This corresponds to the view in the MVC template. It can be used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC.

### *ReactJS Features:*

React.js properties includes the following

- React.js is declarative
- React.js is simple
- React.js is component based
- React.js supports server side
- React.js is extensive
- React.js is fast
- React.js is easy to learn

### *JSX:*

In React, instead of using regular JavaScript for templating, it uses JSX. JSX is a simple JavaScript that allows HTML quoting and uses these HTML tag syntax to render subcomponents. HTML syntax is processed into JavaScript calls of React Framework. We can also write in pure old JavaScript.

### *React Native:*

React has native libraries that were announced by Facebook in 2015, which provides the react architecture to native applications like IOS, Android and UPD.

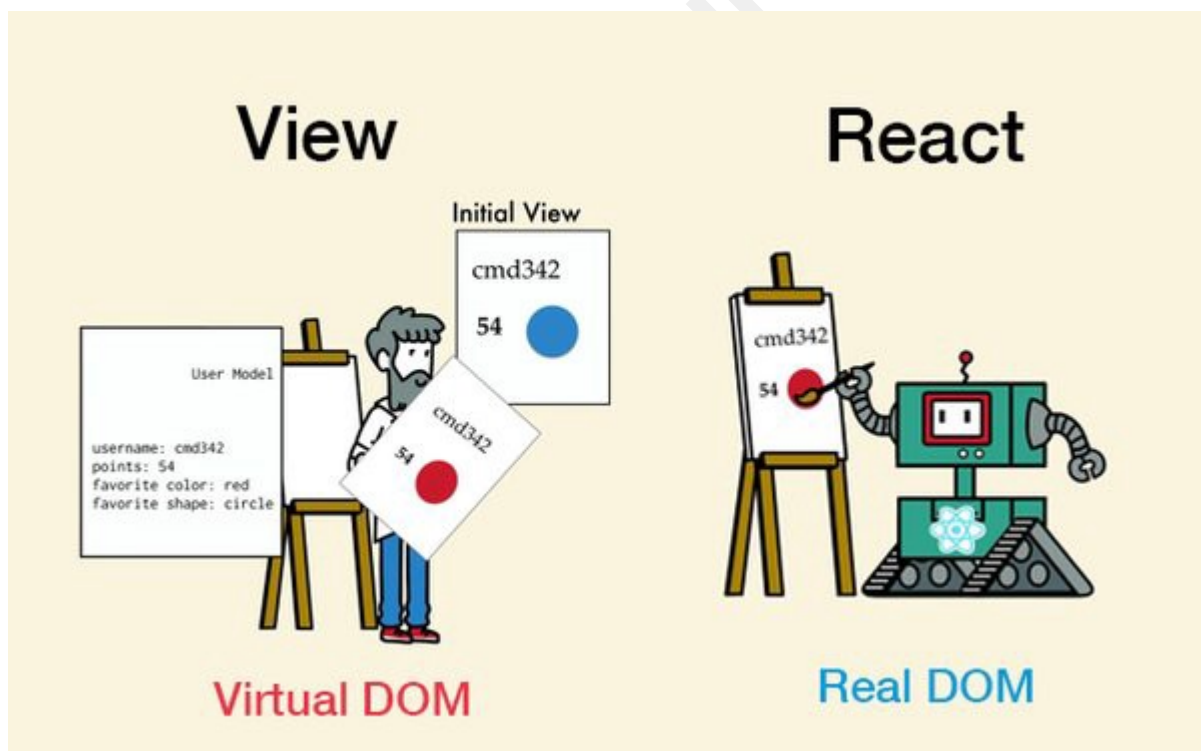
React-native is a mobile apps building framework using only Javascript. It uses the same design as React, letting you utilise/include a rich mobile UI library/ declarative components. It uses the same fundamental UI building blocks as regular iOS and Android apps. The best part of using react-native is to allow/adopt components written in Objective-C, Java, or Swift.

### *Single-way data flow:*

In React, a set of immutable values are passed to the component's renderer as properties in its HTML tags. The component cannot directly modify any properties but can pass a callback function with the help of which we can do modifications. This complete process is known as “properties flow down; actions flow up”.

### *Virtual Document Object Model:*

React creates an in-memory data structure cache which computes the changes made and then updates the browser. This allows a special feature that enables the programmer to code as if the whole page is rendered on each change whereas react library only renders components that actually change.



Source: <https://goo.gl/L7NiIT>

## *Why React?*

The main question arises in front of us is why one should use React. There are so many open-source platforms for making the front-end web application development easier, like Angular. Let us take a quick look at the benefits of React over other competitive technologies or frameworks. With the front-end world-changing on a daily basis, it's hard to devote time to learning a new framework – especially when that framework could ultimately become a dead end.

The popularity of ReactJS as a front-end library can be justified easily by having a look at the features offered by it:

### *1. Simplicity:*

ReactJS is just simpler to grasp right away. The component-based approach, well-defined lifecycle, and use of just plain JavaScript make React very simple to learn, build a professional web (and mobile applications), and support it. React uses a special syntax called JSX which allows you to mix HTML with JavaScript. This is not a requirement. Developers can still write in plain JavaScript but JSX is much easier to use.

### *2. Easy to learn:*

Anyone with a basic previous knowledge in programming can easily understand React while Angular and Ember are referred to as 'Domain-specific Language', implying that it is difficult to learn them. To react, you just need basic knowledge of CSS, HTML and JS.

### *3. Native Approach:*

React can be used to create mobile applications (React Native). And React is a diehard fan of reusability, meaning extensive code reusability is supported. So at the same time, we can make IOS, Android and Web applications.

### *4. Data Binding:*

React uses one-way data binding and an application architecture called Flux controls the flow of data to components through one control point – the dispatcher. It's easier to debug self-contained components of large ReactJS apps.

### *5. Performance:*

React does not offer any concept of a built-in container for dependency. You can use Browserify, Require JS, EcmaScript 6 modules which we can use via Babel, ReactJS-di to inject dependencies automatically.



## 6. Testability:

ReactJS applications are super easy to test. React views can be treated as functions of the state, so we can manipulate with the state we pass to the ReactJS view and take a look at the output and triggered actions, events, functions, etc.

### *Creating a react app:*

There are two ways to create React apps.

First, you use npm (Node Package Manager) installed on your machine. If we're using VS Code, we need to make sure we've configured your machine to run React code in VS code using npm. We will also need to set up a build environment for React that typically involves use of npm (node package manager), webpack, and Babel.

### **ChartJS:**

Chart.js is a free, open-source JavaScript library for data visualisation, which supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. Created by London-based web developer Nick Downie in 2013, now it is maintained by the community and is the second most popular JavaScript charting library on GitHub by the number of stars after D3.js, considered significantly easier to use though less customizable than the latter. Chart.js renders in HTML5 canvas and is widely covered as one of the best data visualisation libraries.

The HTML5 element gives an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.

It is responsive in nature that means it redraws the chart on resizing the window for perfect scale granularity.

This library supports 8 different types of graphs :

- Line
- Bar
- Doughnut
- Pie
- Radar
- Polar Area

→ Bubble

→ Scatter

**Installation:** Latest version of Chart.js can be downloaded from Github or use a Chart.js CDN. Sample Code Snippet of a bar chart plotted in the project:

```
<div className="chart1">
  <div>
    <Line
      data={{
        labels: ["Tex", "Tex", "Tex"],
        datasets: [
          {
            data: [14, 20, 100],
            fill: false,
            borderColor: "#0077b6",
            borderWidth: 1,
            pointRadius: 0,
          },
        ],
      }}
      height={80}
      width={130}
      options={{
        maintainAspectRatio: true,
        scales: {
          yAxes: [
            {
              ticks: {
                beginAtZero: true,
                stepSize: 20,
                fontSize: 8,
              },
            },
          ],
          xAxes: [
            {
              ticks: {
                fontSize: 8,
              },
            },
          ],
        },
        legend: {
          display: false,
        },
      }}
    />
  </div>
</div>
```

**Version Control System:** Version Control Systems are the software tools for tracking/managing all the changes made to the source code during the project development. It keeps a record of every single change made to the code. It also allows us to turn back to the previous version of the code if any mistake is made in the current version. Without a VCS in place, it would not be possible to monitor the development of the project.

### Git and GitHub:

**Git:** Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

**GitHub:** GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

S.No.	Git	GitHub
1.	Git is a software.	GitHub is a service.
2.	Git is a command-line tool	GitHub is a graphical user interface
3.	Git is installed locally on the system	GitHub is hosted on the web
4.	Git is maintained by linux.	GitHub is maintained by microsoft.
5.	Git is focused on version control and code sharing.	GitHub is focused on centralized source code hosting.
6.	Git is a version control system to manage source code history.	GitHub is a hosting service for Git repositories.
7.	Git was first released in 2005.	GitHub was launched in 2008.
8.	Git has no user management feature.	GitHub has built-in user management feature.

## **React Developer Tools :**

React Developer Tools is a Chrome DevTools extension for the open-source React JavaScript library. It allows us to inspect the React component hierarchies in the Chrome Developer Tools.

The Components tab shows us the root React components that were rendered on the page, as well as the subcomponents that they ended up rendering.

By selecting one of the components in the tree, we can inspect and edit its current props and state in the panel on the right. In the breadcrumbs we can inspect the selected component, the component that created it, the component that created that one, and so on.

If we inspect a React element on the page using the regular Elements tab, then switch over to the React tab, that element will be automatically selected in the React tree.

The Profiler tab allows us to record performance information.

This extension requires permissions to access the page's React tree, but it does not transmit any data remotely. It is fully open source, and we can find its source code at <https://github.com/facebook/react/tree/master/packages/react-devtools-extensions>.

## **Pesticide Chrome Extension:**

This extension inserts the Pesticide CSS into the current page, outlining each element to better see placement on the page.

Forked version of Pesticide which removes the black hover bar overlay.

Pesticide is an open source CSS tool created by Adam Morse and available for download, forking on Github at <https://github.com/mrmrs/pesticide>.

## Project Demo :

[Mobility Solutions - Demonstration Video](#)

## Learning Resources:

[The Net Ninja - Full Modern React Tutorial](#)

[React Tutorial For Beginners - Dev Ed](#)

[React - The Complete Guide \(incl Hooks, React Router, Redux\) - Udemy](#)

[Bloggify \(Additional Project Designed during the internship tenure\)](#)

## Conclusion:

It was a wonderful learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the developers' industry.

It was due to this project I came to know how professional software is designed. In a nutshell, this internship has been an excellent and rewarding experience.

Thank you for this amazing experience @Babel Team :)

Regards,

Mr. Saurabh Chidrewar,

B.E. (Pursuing),

Pune Institute of Computer Technology,

Dhankawadi, Pune,

Maharashtra, India - 411 043.