ÖZYEĞİN UNİVERSİTY
FACULTY OF ENGINEERING

CS300

SUMMER PRACTICE REPORT

Mustafa YILDIRIM
S006708

INTERNSHIP COMPANY & DEPARTMENT:

NETAŞ / Multimedia Call Processing Solutions

20.02.2019-05.04.2019

SUMMER PRACTICE REPORT

STUDENT	
Name	**Mustafa Yıldırım**
Internship Start Date	**20.02.2019**
Internship Completion Date	**05.04.2019**
Total Working Days	**20**
COMPANY	
Name	**Netaş**
Department	**Multimedia Call Processing Sol.**
Address	**Yenişehir Mah. Osmanlı Bulvarı No:11 Kurtköy-Pendik/ İstanbul**
SUPERVISOR	
Name	**Çiğdem Vural**
Title	**Team Leader**
Department	**Multimedia Call Processing Sol.**
Phone	
E-Mail	**cvural@netrd.com.tr**
Signature	

DAILY WORK SUMMARY

DAY	DATE	WORK DESCRIPTION
1	**20.02.2019**	In the first day, NETAŞ is introduced. I met my team members in Global Product Support (GPS) department. The department maintains support for products such as KANDYLink, CPaas across the world. Different sections of NETAŞ and the products those are supported are told me detailed by my mentor in the team.
2	**22.02.2019**	Weekly team meeting was organized. I met our team. The team told their problem in detailed that they would like to solve. They make calls or write reports problems on different products with different problem types. The problem was lack of a tool that they cannot find a nice way to filter, search keywords on the pager reports those they are keeping. My duty was to handle this problem by developing a Pager Report Tool.
3	**26.02.2019**	We discussed on Pager Report Tool details. ReactJS framework is decided to complete the project. NodeJS also and MySQL database system were decided as back-end development frameworks and tools.
4	**27.02.2019**	Team Leader (Çiğdem Vural) showed me unwanted fields in existing reporting system. She designed a view UI of new tool to be implemented. I would develop the project according to the view. Our main goal was reducing fields in report tool and add some functionalities to had better search options in the reports.
5	**01.03.2019**	I started to make my research to accomplish this duty. I observed similar projects on the internet. Then, I started to develop UI part of the project by dividing project into sub-modules. The project was available to divide it since it was a web application that consisted of multiple web pages. Every page can be considered as different module roughly.
6	**05.03.2019**	I attended weekly scrum meeting with my mentor (Ömer Kırcalı). We discussed progress so far in the project. I made a small demo on the tool for inserting a new report page. He made some suggestions on different use case scenarios for better user experience. Then I started to enhance the UI.
7	**06.03.2019**	MySQL server environment is installed on the computer. The connection had not been initiated with Pager Report Tool (PRT) yet. On the other hand, the schema of reports written on the server by different tables on database.
8	**08.03.2019**	I tried to insert my first pager report via PRT on the UI by connecting it to MySQL database. Unfortunately, my first attempt failed since direct communication between Nodejs(back-end) and ReactJS(front-end) was not available. I had learned that I should used a RESTful API to connect them. Then, I started to search on REST APIs on ReactJS and NodeJS
9	**12.03.2019**	The day started with weekly scrum meeting. I told my problem on connecting front-end and back-end parts with an API. Then, he suggested me to get help from one of colleagues in GPS. He had done a project on ReactJS with RESTful API. I met my colleague to get information on the subject. According to his advices, I installed Axios package on my development environment as http client to make connection NodeJS server.
10	**13.03.2019**	I continued my work on solving API problem to insert pager report data to the database. The problem solved by using axios. I tried to load data with different types to test whether it was correctly inserted to the database. I could not find any problems according my observations. Then, I worked on UI to enhance UI and UX.

Student's Name: **Supervisor's Name:**

Student's Signature: **Supervisor's Signature-Stamp:**

DAILY WORK SUMMARY

DAY	DATE	WORK DESCRIPTION
11	**15.03.2019**	I made a demo for Insert report module – the module was completed until that moment- I showed how to insert a report to the system in front of whole team. Then I got some feedbacks from the team. The first concern was the UI since it was not user friendly. I had given some advices to improve the UI and some of report field were reduced according team needs.
12	**19.03.2019**	I started my research to make UI better. The UI must be neat and fully functional. I thought that I must use an existing library to make UI better. Then, I installed react-bootstrap.js NPM package. It gave me an infrastructure to make a simple UI with less code work.
13	**20.03.2019**	The UI is enhanced by react-bootstrap. Then, I started to implement Search Report module. The module was designed to search the reports by keywords, problem types, product types and filter it by date. To do that, I used a package react-bootstrap-table2.
14	**22.03.2019**	I started to implement related changes for search module on server side and API. I had some problems while retrieving data from database. The problem was about asynchronous methods on APIs. I had to waste some time to understand origin of the problem.
15	**26.03.2019**	We did scrum meeting and my mentor found a bug while inserting data. If the report was inserted with text including single quotes in it, SQL query wouldn't work. I have used regex to solve the problem. Mentor wanted the app with a login facility so that app would be used by only registered users.
16	**27.03.2019**	I made a login page as a module in the application. I made related arrangements in database and NodeJS server. User table is added to database to register users.
17	**29.03.2019**	Team leader wanted me to implement an admin functionality for the system. Admin could have flexibility to customize problem types, product types, problem subfields, adding/deleting users.
18	**02.04.2019**	According to the feedback in previous meeting I started to apply related changes in database, server and API calls in the UI.
19	**03.04.2019**	The third module was implemented for administration as team leader wanted. The module was Control Panel. The panel was only open for admins.
20	**05.04.2019**	I demonstrated last situation to the team. We discussed how improvements could be done. I continued developing control panel. I also fixed some bugs on logging in as user or admin. There were some problems to check users from database.

Student's Name: **Supervisor's Name:**

Student's Signature: **Supervisor's Signature:**

Abstract

I have completed my first internship in NETAŞ Multimedia Call Solutions Department. Our department handled support product those are developed by NETAŞ or some other products. Also, the department deals with fixing and reporting bugs after support. However, the team that I involved was dealing with only supporting to the related product. Then, they report the problem to GTS team to fix bugs, if they could not solve the problem. The reports are kept in any case regardless whether problem is solved or not. My duty was bringing a web application that simplify existing reporting tool with reduced fields. The tool can be seen in Appx. 1. As it can be seen, there are too many fields to fill. Our team wanted to search and filter old reports to get an idea by analyzing other colleagues' reports. Therefore, they needed simplified version of the tool. Besides that, the tool had not a search option for a specific keyword or a date filter. Also, the team thought that, a web application that runs on browsers can be more useful rather than a desktop application.

In light of the information above, I have developed my existing skills on RESTful APIs along the development. On the other hand, I almost learnt all of my JavaScript skills in the internship. JavaScript environment is broad to explore. Nevertheless, I was able to finish my project by exploring some packages via NPM which a JavaScript package installer for NodeJS environment is. As a result, this internship improved my skills on JavaScript, web development, RESTful APIs.

I. Introduction

I developed a pager report tool as requested from my team leader. The development of project started with a brainstorm on implementation of Pager Report Tool. The main problem was that old-fashioned report tool (Appx. 1) does not satisfy needs of the team in order to analyze older solution of same type problems in reports. For instance, a team member got a call from AT&T for product KandyLink he could not solve the problem. Then, he wanted to take a look at previous reports to search same kind of problem to come over existing problem. However, our team member suffered a lot due to Ribbons old report tool since they could not group and view reports by filters like problem type, product, customer, supporter. We thought that a web application that can fulfil these needs can reduce time consuming remarkably. I was more experienced on Java environment until that moment. Then I went back to continue my research on ReactJS and necessary tools for it as we agreed with team leader before. As a result, I got familiar with Microsoft VS Code as code editor instead of Eclipse for java. I started to implement early stages of UI in first week of the internship. UI implementation had to completed with database. We have chosen MySQL RDBMS for this purpose since our team has experience on it and there are a lot of resources with ReactJS. I had emphasized RESTful API work flow in next stages to make an interface between NodeJS server and UI(ReactJS). I spent a long time to learn this and fix the problems along the development cycle. Development cycle includes weekly scrum meetings my mentor and occasional meetings with team. I accomplished core part of PRT at the end of 20 days of internship period with some bugs.

II. Company Description

Netas is mainly a telecommunication company that brings solutions to problem of other companies or individual customers. The company is established in 1967 to bring telecommunication solutions to Turkey with Canadian partner Nortel. Also, Netas defines himself as a system integrator. The reason for this definition is that Netas provides services and products for companies in a wide spectrum. The spectrum can be understood via Netas services and products. For instance, Netas has a collaboration with Aselsan on ULAK 4.5G GSM project. It can be said that it is an telecommunication project area. However, Nova Cyber Security project can be given as a different field from telecommunication. These various type of projects from different fields prove that Netas is a system integrator as it defines itself.

I worked in Multimedia Call Processing Solutions department which is under International R&D department in Netas. Our department was focused on giving support for products of Netas in telecommunication field. For instance, Netas provides a software named KandyLink for Ribbon Telecom Company. KandyLink or SPiDR(new name of KandyLink) is real time communication software for individual users or other companies. This product is just one of the products that our department supports. Our department gives the support via bug fixing, testing the software or calls from customers. Besides that, our team was in GPS(Global Product Support) which gives support for the companies. Our team considers incoming problems to solve them. Our team keeps record of those problems as problem reports. There were a lot of unrelated fields in existing report tool as our team told me. Our team needed a solution via a new report tool with reduced fields in order to view source of problem and solution method in previous problems.

My role was described as bringing a new Pager Report Tool(PRT) in order to fulfill new needs of our team. The software that was used in the project decided on together.

III. Pager Report Tool (PRT) Renewal

i. Problem Statement

This project is designed to bring a solution for previous Pager Report Tool Appx2. The first problem was that the previous tool was not a single tool that designed specifically to reporting problems. The tool was a sub-component of TechOps. Therefore, it was not convenient in order to maintain all needs of our team. This was causing mess for our team. Also, there were additional functionalities which are not used regularly. Besides that, some other functionalities requested based on new needs. Then, the previous tool has old-fashioned UI design. Our solution should have been under some of constraints based on new needs. The constraints are as follows:

- The tool must come with reduced functionalities
- The UI must be clearer to search reports
- Viewed reports must be editable with less effort
- Admin and standard user login feature must be provided

The problem was a new problem for our company specifically for PRT. However, there were some other tools which are needed to maintain is reviewed. The solution is brought by considering other solutions for other tools used in the company.

ii. Tools and Techniques Used

We have reached a consensus on programming languages, software, hardware and techniques after discussions with the team. Our decisions as follows for each them:

- Programming languages: JavaScript [1] is decided as the language since the project is requested as a web application. JavaScript was the best choice for a user-friendly UI design. The language gives developers flexibility to deal with HTML pages and components. It is chosen also for back-end design. The reason is that, there was a nice runtime environment named as NodeJS for back-end purposes. Thus, we would not have to deal with incompatibilities between back-end and front-end for a small project with a limited time. The alternative can bee seen as JavaEE[2] with its components such as JPA, JSF,EJB. Java was not a suitable choice since it requires huge prior knowledge and it was commonly used for large-scale web projects. Therefore, JavaScript is chosen with benefits as fast development process, less prior knowledge requirement for the environment, large community support for web development-especially for front-end.

- Software: ReactJS [3] view framework is chosen for front-end development. The prominent reason for that was its component-based structure. The structure helps us to develop each part of project separately. This approach makes us proceed step by step and see the results for each part separately. Also, it prevents contagion of a single failure to all parts of tool. Development time is reduced by the framework. There are a lot of alternatives such as VueJS, AngularJS. However, none of them has a large community support as ReactJS. NodeJS [4] is runtime environment is used for back-end purposes with its lightweight structure and compatibility with ReactJS due to JavaScript. The environment gives us ability of easy setup of a server with a few lines of code. Unlike java backend environment (EJBs,Application Servers or Web Servers), it handles communication between database server and ReactJS only by using RESTful API [5]. Thus, we gained simplicity in development process. This helps us to detect our problems easily and solve them. NodeJS environment is chosen over JavaEE since the second option is commonly used for large-scale project unlike our lightweight web application. If the second option was chosen, the production and deployment process would be painful more than JavaScript environment. The other beneficial part is that we were able to add needed packages via NPM [6] with just a command to our project. There are a lot of packages in our project. However, there is a prominent package of NPM makes PRT enough for team needs. React-bootstrap-table2 [7] is the package that makes the reports easily readable and editable as the team wants. Apart from those packages, we had to use an IDE in order to write our code. VSCode[8] is chosen for this purpose since it was suitable for JavaScript development. Large amount of JavaScript developers use VSCode for development. Thus, we would able develop our project fast.

- Hardware: In first stage, our plan was deploying our application on one of Netas servers. However, we could not deploy it on the server due to technical problems on the server. The deadline was so close after production of the software. Then, we decided to use a PC that I have used in my internship for hosting the web application.

The PC has enough specification for use of our team with 8 GB of RAM and Intel i5 CPU.

- Techniques: We have used MERN [9] stack structure with a little change by using MySQL database instead of MongoDB in development. The technique provides us a pattern in order to combine database server, REST API, ReactJS and NodeJS. In this technique, NodeJS server communicates with database. REST API makes a bridge between front-end (ReactJS) and back-end (NodeJS). First, we have chosen MySQL instead of MongoDB since we do not really need a No-SQL database. We do not have large amount of data transaction while MongoDB is used for scalable database structures. The second, I can get support easily from the employees since they have knowledge on mostly MySQL. MERN stack technique is a simple technique for CRUD (Create, Update, Delete, Read) operations. PRT simply does nothing other than CRUD operations. Therefore, the technique fits our needs and development process. Another benefit is that, there are a lot of resources about this technique. As a result of this, we were able to find solutions about our problems on CRUD operations immediately.

iii. Detailed Explanation

As it is mentioned in part ii. I have used JavaScript, indeed ReactJS is used mostly for development. The development is progressed via VSCode. Overview of the project's components can be seen in Appx. 3. Our main concern was deploying this project within a limited time to fulfill PRT requirements. Therefore, we have chosen tool and techniques described in part ii. according to this purpose. Component based structure of ReactJS is utilized in the project for UI part. As an example, Login.js component is written for login page Appx 4. According to our needs other components are implemented for UI part of the project. Some packages were needed to add to the project via NPM. For instance, react-bootstrap package was suitable for beautifying UI. The package is added to project folder via `npm i react-bootstrap` command. This is how NPM works also for other package with their NPM commands. Other main components of PRT(InsertReport, SearchReport, Control Panel) are implemented by this way. This technique is used for core part of UI development. The UI part has changed many times along development period in order to gain better UX (user experience) according to feedbacks given by team members. The project must be served on server. NodeJS was used for this purpose. PRT can be served on http://localhost:3000 address.

As it is has been said that PRT is designed based on components. Each component has implemented according to their functionality.

Component	Permission	Sub-Component	Functionality
Login	Admin/User	non available	Used for opening sessions
InsertReport	Admin/User	avaliable	Inserting reports page
SearchReport	Admin/User	non avaliable	Search and edit reports
ControlPanel	Admin	avaliable	Administration panel

PRT mainly consists of single components for each page. Implementing sub-components was mandatory for some pages such as ControlPanel since functionality of components must be divided. The division of components decided on transactional relation of components with back-end side. The division operation must be implemented to avoid single point of failure risk. A single failure due to database relation may affect whole page. ControlPanel is divided four sub-components such as Product&Problem Insert, Remove Problem&Product, Add User, View User Appx. 5. The UI part is accomplished with the perspective described above. Our main concern was reaching modularity in order to detect bugs, proceed step by step, gathering fast feedback for each phase of implementation.

Apart from implementation of UI. The second and most challenging phase was combining this structure of UI with backend side. MERN stack is used by our way of design and implementation as it is mentioned in part ii. MERN stack technique is commonly used among developers with MongoDB database server. We have chosen MySQL databse server instead of that due to reasons mentioned in part ii. The structure gives us ability of modular development and testing each module separately. This can be considered as sort of divide and conquer approach in terms of design of a software. The technique can be understood properly with the figure below:

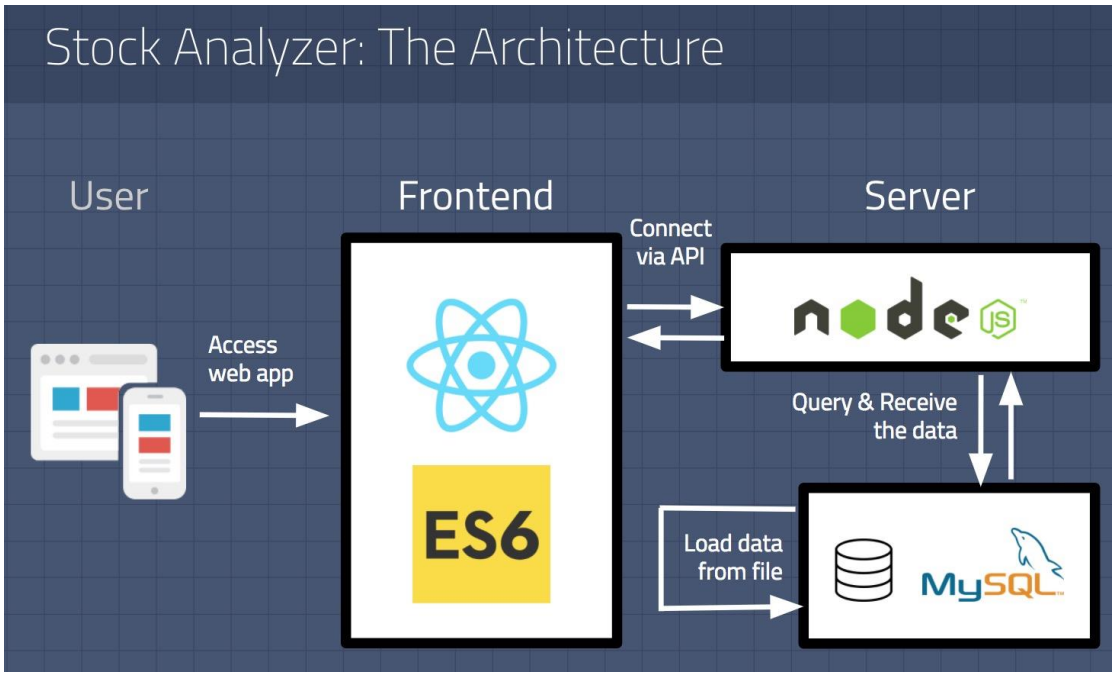


Figure 1MERN STACK

Front-end implementation is discussed before. The right side of the figure should be discussed now. We have implemented our UI design and implementation. According to that, communication with database must be implemented. MERN stack provides an easy path to do this. Firstly, I handled communication between MySQL database and NodeJS server. I have tried to do basic CRUD operations via manipulating server.js file which runs NodeJS server. I have tried to do it with some fake data to see whether NodeJS and MySQL communicates well or not. I have monitored the results via MySQL WorkBench application Appx 6. The point is that, I make myself sure that each module connection is prepared well. MERN technique was quite convenient to test each connection of the components. Thus, it reveals that the technique that is used was successful for the purpose. Then, REST API part is checked to see that the integration is completed. REST represents for Relational State

Transfer. The meaning of this concept can be perceived via its name. We could be able to transfer data from NodeJS to ReactJS or vice-versa via this approach. JSON which stands for JavaScript Object Notation file structure was used to reach this goal. After the implementation of each module. The integration of components was tested with the team members. Team members have found some bugs on data transfer between components such as data transfer with single quote character. The data transfer was not successful with single quote due to characteristic of SQL. I had to fix these kinds of bugs after integration tests with feedbacks of the team.

We have discussed where to deploy the app after the integration of the app. However, the first part of the internship part is end. I have completed this requirement in the second part of my internship.

iv. Results

I have completed first part of my internship. The solution that I have provided to team is accepted mostly by our supervisor and coworkers. The reason is that I have solved most of the problems of old PRT. I have reduced fields of pager reports with feedbacks of team members in regular SCRUM meetings. The most challenging part was bringing a table structure which is editable and readable easily.

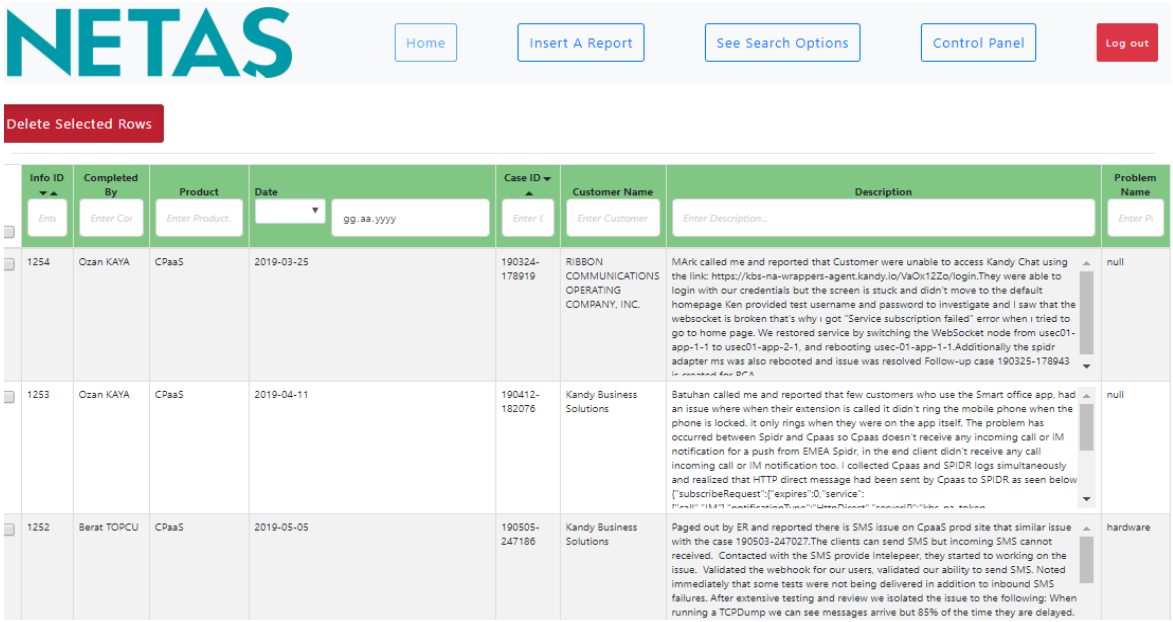


Figure 2 SearchPanel and Table view

If Figure 2 and Appx 7 is compared, the success of the new PRT can be seen by editable and readable new PRT. The most prominent difference is that, users can view multiple reports in just one page. Also, they can choose number of reports they would like to see for each page. This solution brings the team a lot of flexibility to read old pager reports. Thus, they can analyze old reports easily to have an idea about ongoing problems those they have faced. Secondly, they can edit multiple fields or they can apply many filters such as date, sorting by names or case IDs etc. The tool solves older PRT problems with my approach. The tool can enhance teams' speed while they seek some solutions for new problems by scrolling older pager reports easily and fast.

The second beneficial feature was on UI and UX. PRT shows that the new approach was helpful about graphical interface. The colorful background of component was helpful to read the reports properly. They had to click on new pages to see other reports in previous tool. As they have told, the new tool is quite successful to view and edit pager reports. This shows us that user experience of PRT is enhanced. Apart from that, our supervisor made some criticism about other components of our tools such as InsertReport component since it was not as good as table view. The problem was that the date entry of the tool was not resizable Appx 8. I have worked on the problem, but I could not solve it. Also, the problem was not a prior problem as they said. As a result of that, I have focused on other problems of core functionalities of the tool.

We had some concerns before implementing the problem of latency while retrieving the reports from database. We did not have a chance to load enough data to database to test the system in my first part of internship.

IV. Conclusions

I have completed first part of my internship with experience by participating a real work environment. The education that I have get from my school helped me a lot in some critical points of development a software. I have gained basic principle of object-oriented programming (OOP). OOP lightened my perspective to develop a modular system. This perspective is quite important especially where people works with teams. In addition to that, CS222 lecture made my code clear since the lecture taught us how to make code clean and readable. Another big support for my work come from CS202 database systems lecture in order to solve database problems and understand relational database structure.

If we compare class assignments and internship projects, they are completely different in way of development cycle. The requirements of the project was changed every week by our team. This experience taught me adopting the struggles of development cycle.

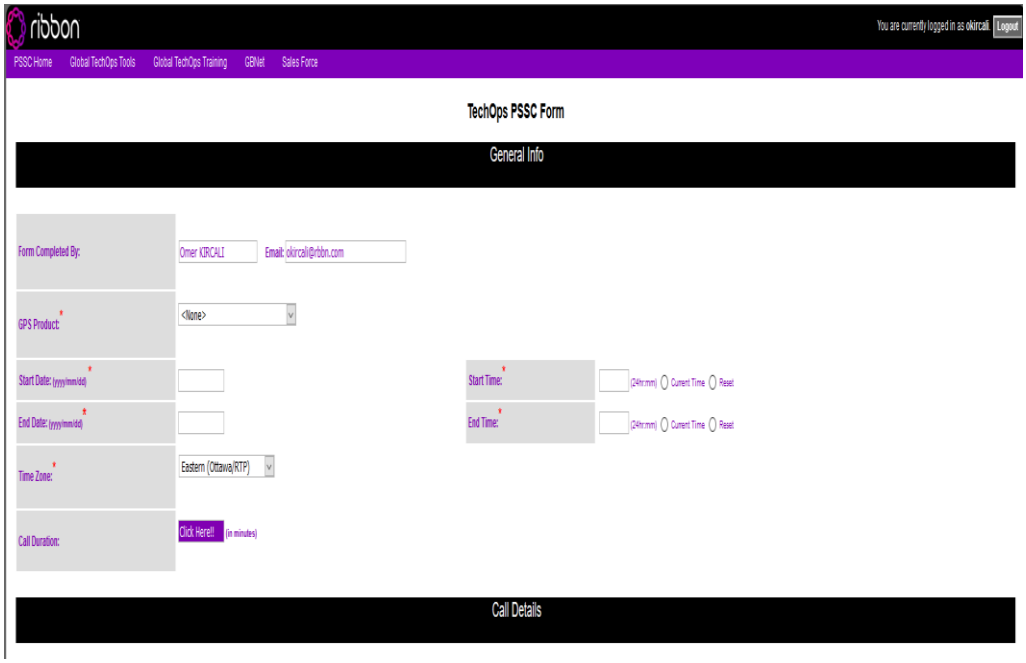
I have learned REST API structure during the internship. The structure showed me that there was no direct communication between components in real software structures. The communication was handled mostly with interfaces.

I had been thinking about being full-stack web developer before the internship. I had eager to do this along my internship period. Although, I have faced struggles during the development, it increased my curiosity about web development and JavaScript.

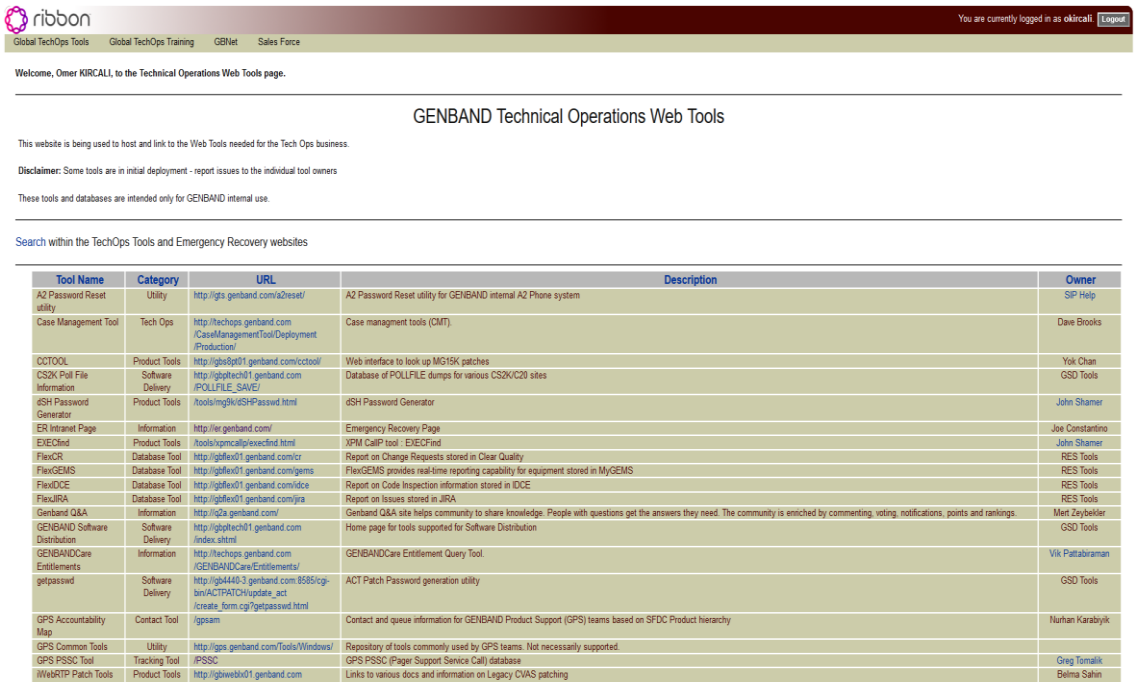
The first phase of my internship is finished with good impressions about the company. The company has flexible work environment and hours. This makes me to learn without forced by other co-workers. I have gained excitement to what I would like to with some constraints.

Appendix

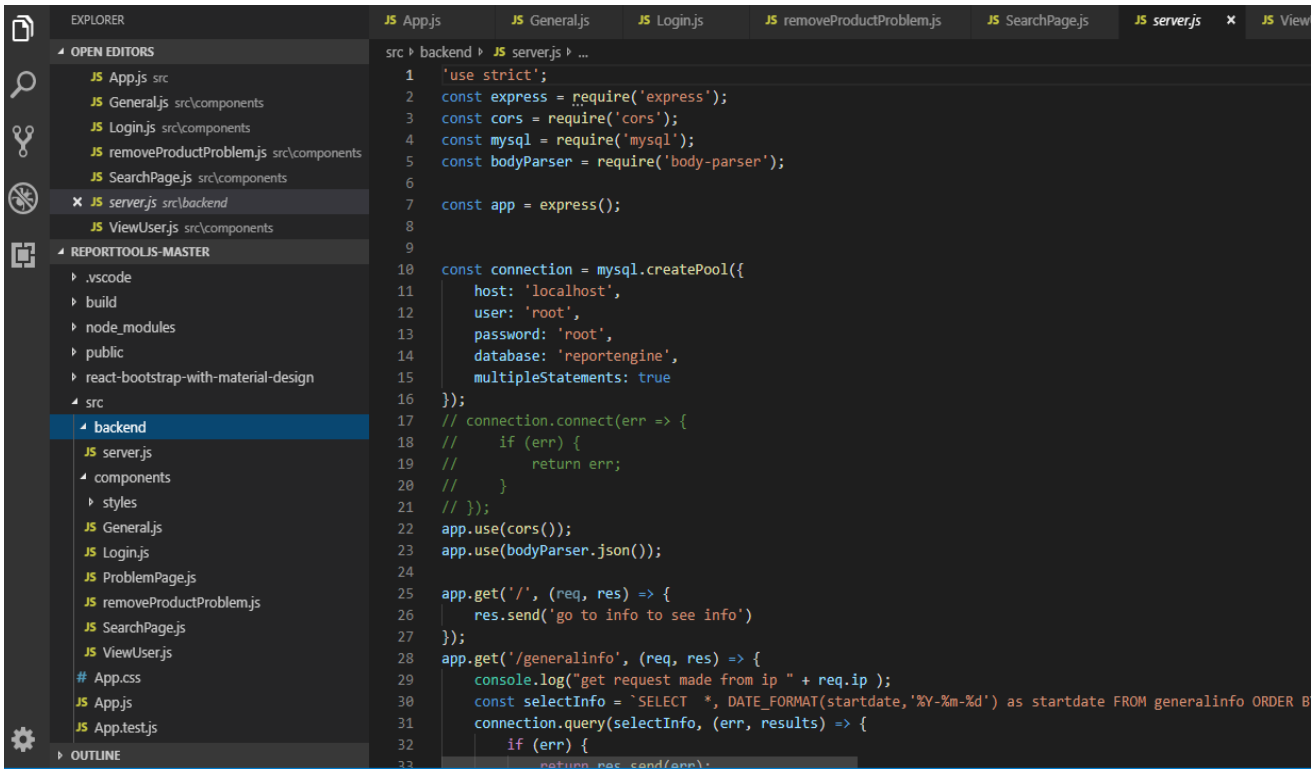
1. TechOps Report Tool



2. Previous PRT as a sub-component



3. Projects main folder in VSCode



4. Login.js and corresponding Login component

App.js | General.js | Login.js | removeProductProblem.js | SearchPage.js | server.j

src ▸ components ▸ Login.js ▸ ...

```
1   import React, { Component } from 'react';
2   import {Container} from 'react-bootstrap';
3   import { Button,ButtonGroup } from 'react-bootstrap';
4   import { ButtonToolbar } from 'react-bootstrap';
5   import Navbar from 'react-bootstrap/Navbar'
6   import { Form, Row,Col,Dropdown,DropdownButton } from 'react-bootstrap';
7   import ViewUser from './ViewUser.js';
8   import RemoveProductProblem from './removeProductProblem.js';
9   const axios = require('axios');
10  var host = window.self.location.hostname;
11  class Login extends Component {
12      constructor(props){
13       super(props)
14       this.state = {
15         currentPage :<ViewUser ></ViewUser>,
16         newUserName: '',
17         newUserPassword: '',
18         newProduct:'',
19         newProblem: '',
20         problemfields: '',
21         pageName:'View Users'
22
23       }
24       this.handleUserInsert = this.handleUserInsert.bind(this);
25       this.handleUserChange = this.handleUserChange.bind(this);
26       this.handleChange = this.handleChange.bind(this);
27       this.handleControl = this.handleControl.bind(this);
28       this.handleProblemInsert = this.handleProblemInsert.bind(this);
29       this.handleClick = this.handleClick.bind(this);
30      }
31      handleClick(event){
32       if(event.target.name == "ppinsert" ){
```

Figure 3 login.js

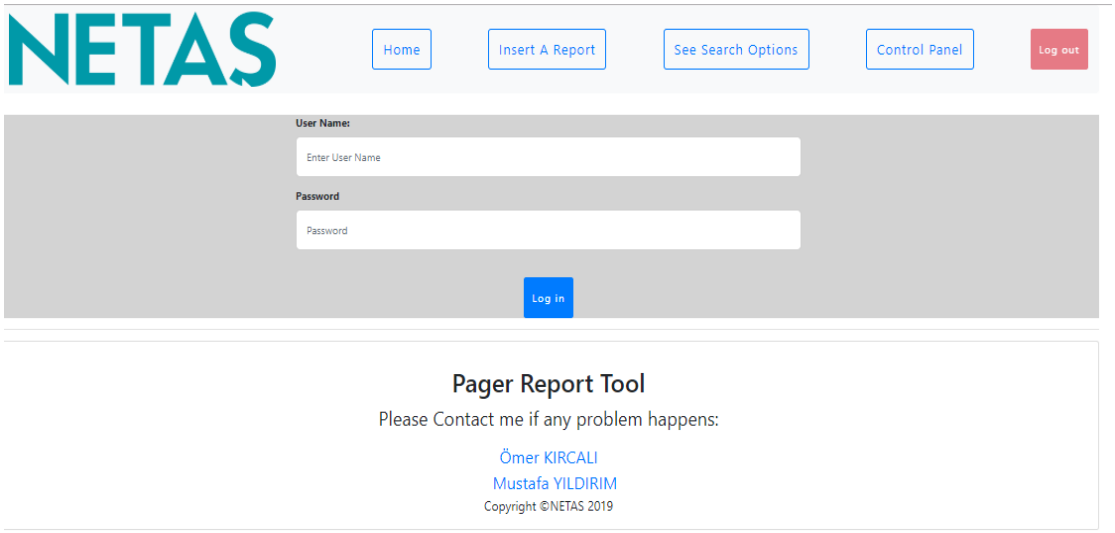
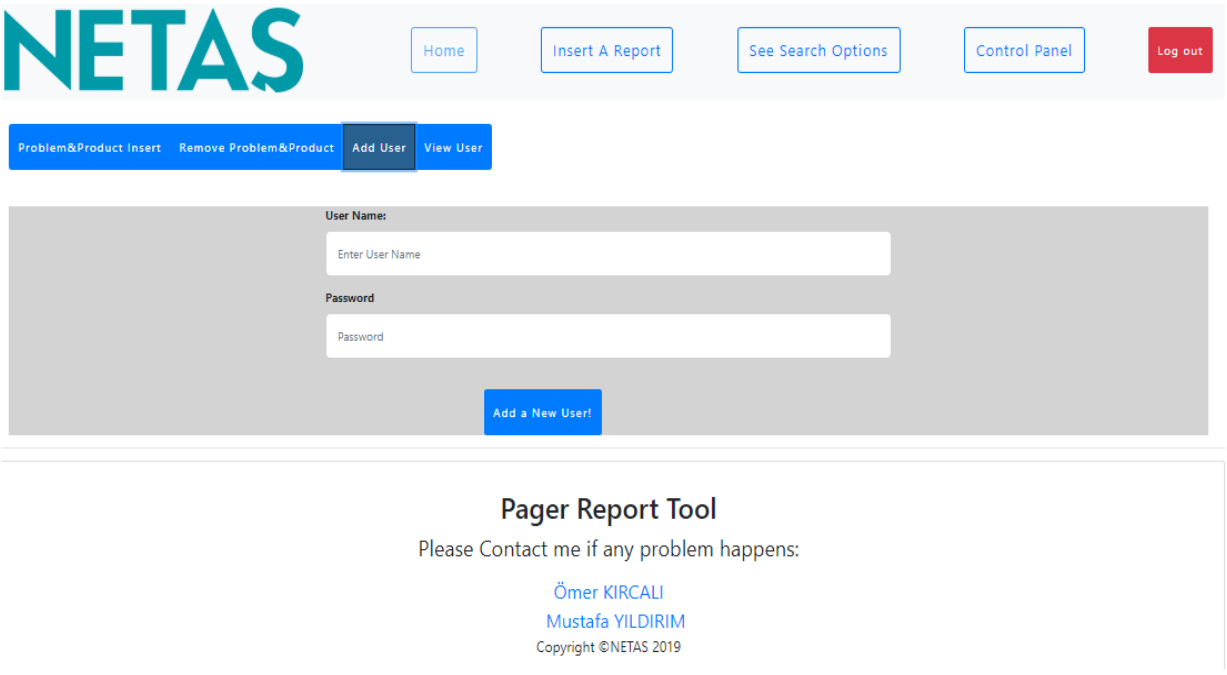
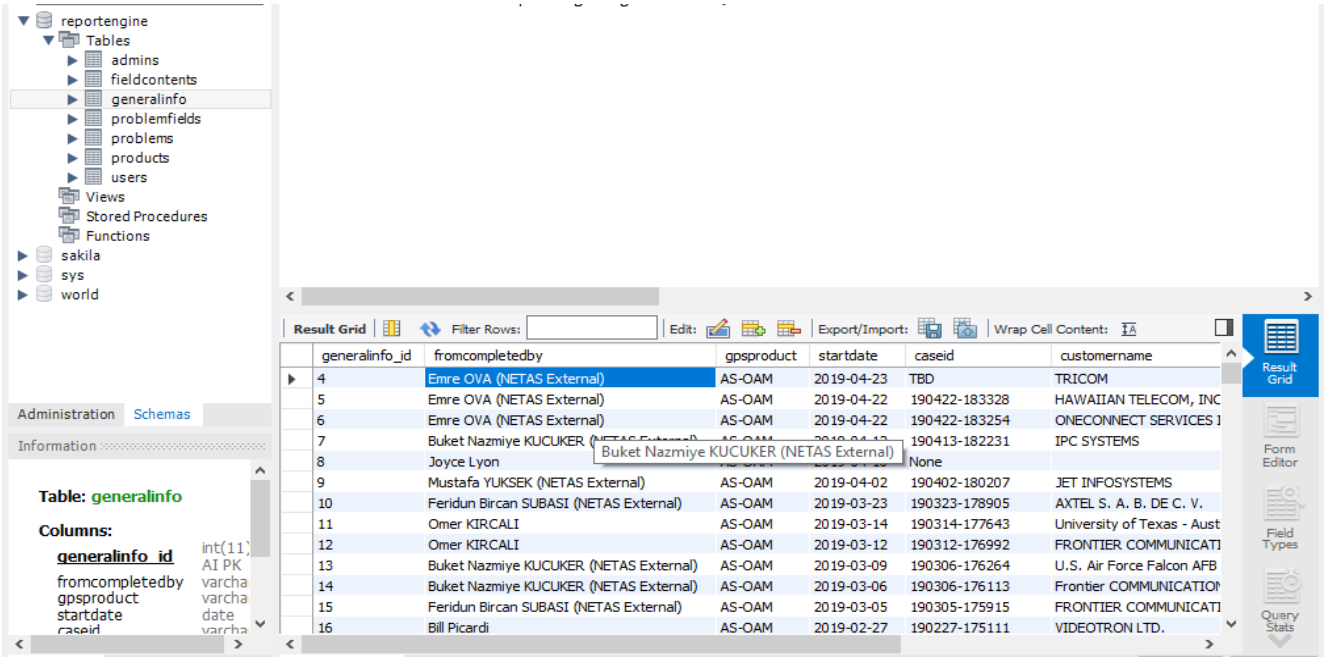


Figure 4 login component

5. Sub-components

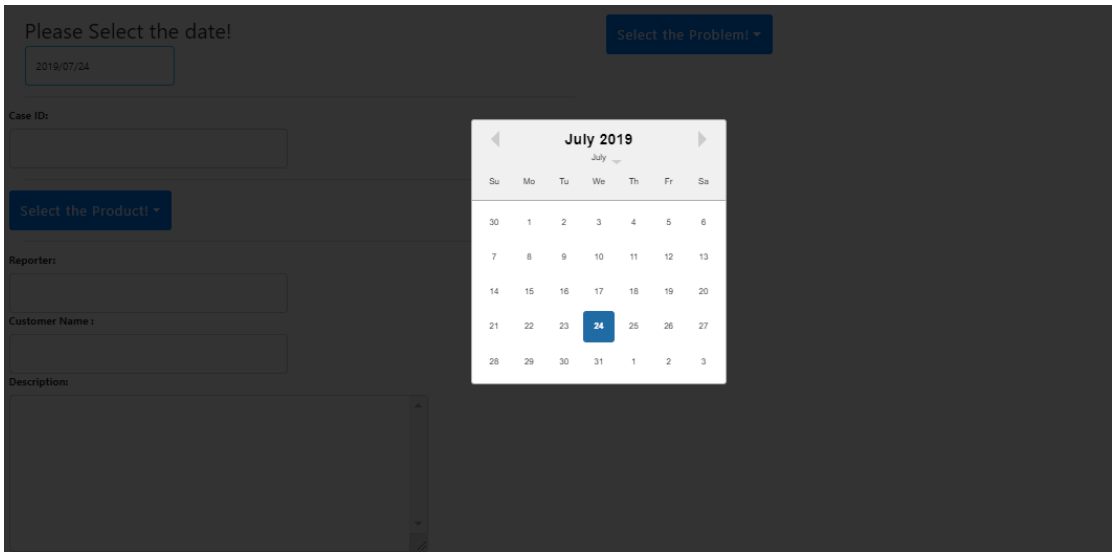


6. MySQL WorkBench



7 Old PRT page

8 Date Entry Modal



References

[1] https://www.javascript.com/
[2] https://www.oracle.com/technetwork/java/javaee/overview/index.html
[3] https://reactjs.org/
[4] https://nodejs.org/en/
[5] https://restfulapi.net/
[6] https://www.npmjs.com/
[7] https://github.com/react-bootstrap-table/react-bootstrap-table2
[8] https://code.visualstudio.com/
[9] https://medium.com/codingthesmartway-com-blog/the-mern-stack-tutorial-building-a-react-crud-application-from-start-to-finish-part-2-637f337e5d61