**DEVELOPMENT SUPPORT TOOL**

**eClerx**

**A training report**

Submitted in partial fulfillment of the requirements for the award of degree of

# B.Tech

# (CSE) Submitted to

**LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB**



**From 05/07/19 to 12/12/19 SUBMITTED BY**

## Name of student: Prerna Bhatia Submitted to:

**Registration Number:** 11611526 **Name of Supervisor:** Rahul Vanpully

**Signature of the student: Designation:** Associate Program Manager

**To whom so ever it may concern**

I, **PRERNA BHATIA, 11611526** hereby declare that the work done by me on “**Development Support Tool**” from **August, 2019** to **December, 2019**, under the supervision of **Rahul Vanpully**, **Designation, eClerx Service Limited,** and **Gaurav Pushkarna**, **HOD,** Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **B.Tech.**

PRERNA BHATIA (11611526)

Signature of the student

Dated:

**To whom so ever it may concern**

This is to certify that **Prerna Bhatia, 11611526** from Lovely Professional University, Phagwara, Punjab, has worked as a trainee in **eClerx Services Limited** on “**Development Support Tool**” under my supervision from **August, 2019** to **December, 2019**. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial fulfillment of the requirements for the award of the degree, degree name.

Name of External Supervisor Name of Internal Supervisor

Designation of the External Supervisor Designation of the Internal Supervisor

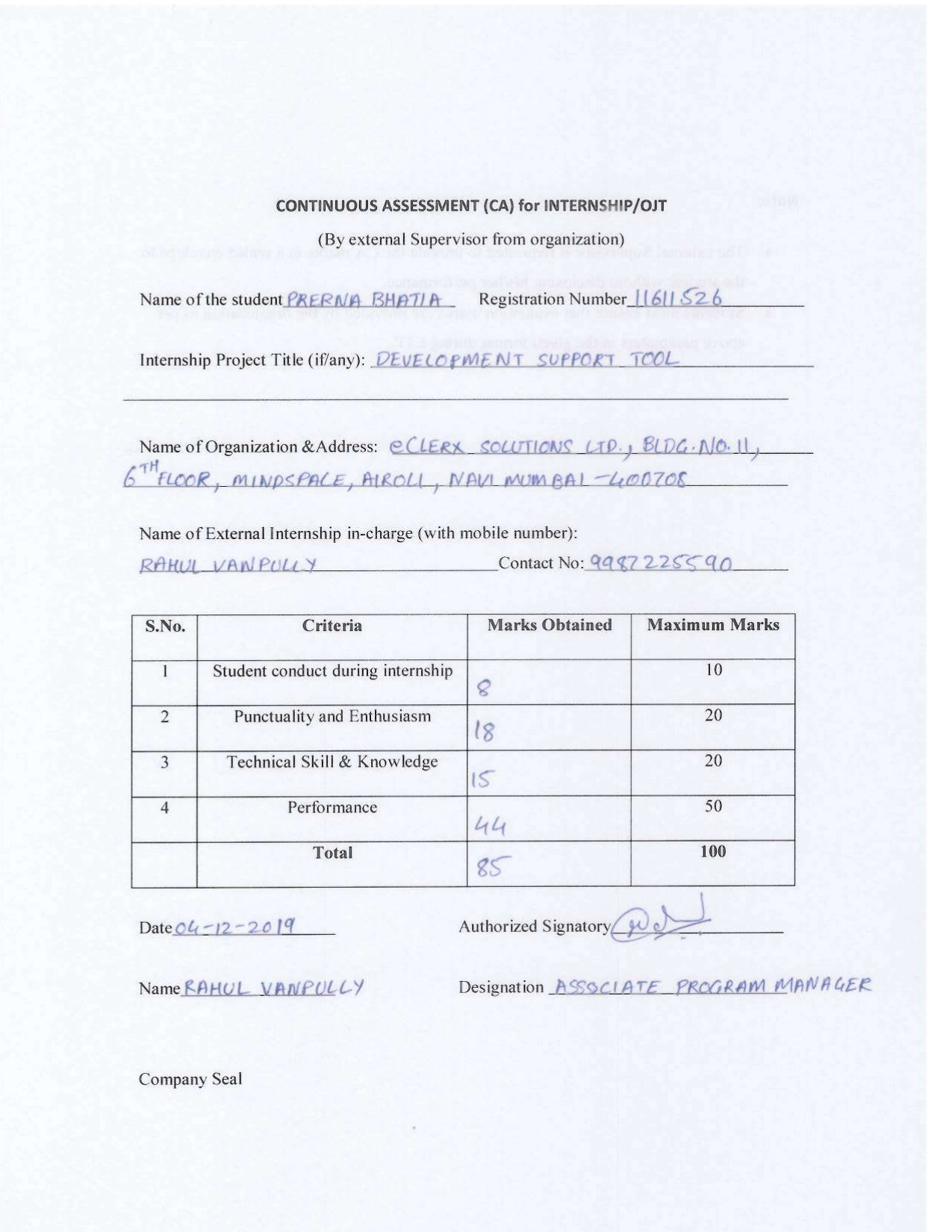
Signature of the external Supervisor Signature of the Internal Supervisor

Dated: Dated:

**Training Letter from eClerx Services Limited**



**MARKS GIVEN BY EXTERNAL SUPERVISOR**



**ACKNOWLEGMENT**

I take this opportunity to express our gratitude and respect to all those who have helped me throughout our working period on the real time company environment. Doing internship in eClerx help us a lot to understand the new technology and how to grow in a corporate world. My special thanks is to our mentor Mr. Rahul Vanpully (Associate Process Manager), who helped me a lot to show us the right path how to work in a company and to Learn the various aspects of application.

I owe my regards to the entire faculty of the department of Computer Science at LPU from where I learnt the basics of Computer Science and I express my sincere thanks to all our course mates who supported us in the project through various informal discussions which were very valuable to the successful completion of the project.

**-PRERNA BHATIA**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Title** | **Page** |
| 1. | Declaration by Student | 2 |
| 2. | Declaration by Supervisor | 3 |
| 3. | Training Certification from organization | 4 |
| 4. | CA Marks by External Supervisor | 5 |
| 5. | Acknowledgement | 6 |
| 6. | List of Tables | 8 |
| 7. | Chapter-1 INTRODUCTION OF THE COMPANY | 18 |
| 8. | Chapter-2 FRONT-END WORK | 28 |
| 9. | Chapter-3 DEVELOPMENT SUPPORT TOOL | 40 |
| 10. | Chapter-4 KOA.JS AND MONGODB |  |
| 11. | Final Chapter- CONCLUSION AND FUTURE PRESPECTIVE |  |
|  | References |  |

**INTRODUCTION OF THE COMPANY**

* 1. **Company’s Vision and Mission**
* eClerx has established a robust, technologically advanced Research and Development center located in Mumbai, India. The center is comprised of a state of the art working lab and includes innovative technologies that help accelerate the design and development of our extensive library of cutting-edge solutions. Several of the outstanding technologies utilized in this R&D center has been instrumental in creating eClerx products.
* Big-data, Machine Learning, Robotics Process Automation, Web Data Harvesting and Document Data Extraction are examples of areas that eClerx has invested in. The lab also boasts fully modernized computers, an array of software, dedicated servers and significant storage space for hosting and testing various eClerx solutions.
  1. **Origin and growth of company**
* EClerx Services Limited is engaged in providing Knowledge Process Outsourcing (KPO) services to global companies. The company provides data management analytics solutions and process outsourcing services to a host of global clients through a network of multiple locations in India and abroad.
* The Company was originally incorporated on March 24 2000 as eClerx Services Private Limited. In August 2007 the Company was converted to a public limited company and the name was changed into eClerx Services Limited. EClerx Services completed its Initial Public Offer (IPO) and the equity shares were listed on the National Stock Exchange of India Ltd. (NSE) and the Bombay Stock Exchange Ltd. (BSE) effective December 31 2007.
* During the financial year ended 31 March 2010 which marked completion of 10 years by eClerx Services the company recorded a turnover of more than Rs. 2500 million for the first time in its history.

.

**Various departments and their functions**

**eClerx DIGITAL**

* eClerx Digital is the trusted partner of choice to the world’s largest global brands for creative production, eCommerce / web operations, and analytics & insights services. We improve profitability for their digital businesses.
* Develop products and platforms that cater to next generation market needs driven by changing global trends in the financial services, cable and telecommunications, retail, fashion, media and entertainment, manufacturing, travel and leisure, software and high-tech industries.

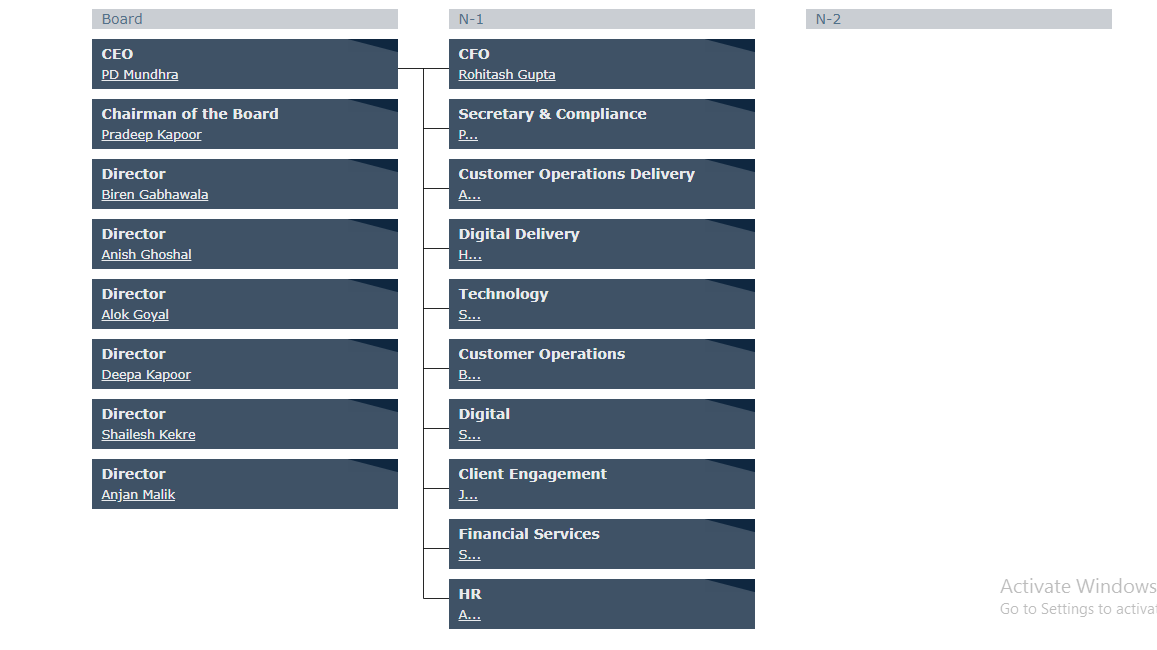
**eClerx CUSTOMER OPERATIONS**

Our multi-disciplinary experts are committed to our clients for the duration of their engagement. Each diverse team of specialists is led by a forward-thinking eClerx Customer Operations Product Manager dedicated to improving customer experience and operational efficiencies. We operate as an extension of your company.

**eClerx FINANCIAL MARKET**

For financial organizations across the world, eClerx Markets, offers consulting, technological innovation, and process management expertise to uniquely solve operational challenges. With nearly two decades of industry experience complemented by the application of smart automation and robotics, our team of experts deliver holistic solutions across the trade life cycle, change management, data analytics, compliance, cash securities operations, document digitization and generation, and outreach.

**Organization chart of the company**

****

**INTRODUCTION OF THE PROJECTS UNDERTAKEN**

**CHAPTER 1: FRONT END DESIGN**

* 1. **Objectives of the work undertaken**
* We are required to design the webpages of the website. I was supposed to design a particular module from the website. I learned bootstrap to make it responsive so that the website can be accessed through any internet connected device and it will show same results every time.
* It is typically measured as a percentage of website visitors who become paying customers. This goal is suited best for e-commerce websites that want to focus specifically on maximizing a website’s ability to sell. This goal can also help measure changes in the quality of traffic.
  1. **Scope of the Work**
* This is arguably the most popular goal for business websites that have lead generation as their main purpose. This goal is best for companies that want to get more sales leads through improving their website’s marketing performance. To measure lead increase, I recommend setting a percentage rate increase as opposed to an absolute lead count. This accounts for monthly traffic fluctuations and allows for meaningful historical comparisons.
* This goal lends to measuring how effective your website is in educating users about your new products or services when you are not necessarily “selling” online. This works well for websites that serve an informational purpose but don’t necessarily need to generate leads or capture information, such as: micro-sites for new product launches, political and government websites or non-profit websites dedicated to raising awareness around a certain cause.
  1. **Importance and Applicability**
* Creating a website is not just to promote your business, it helps in creating brand awareness, gives a platform for people to interact, approach and discover as well. To market a product or service effectively, you should have a web page that is viewable in a variety of devices. In today’s world you have smart phones, tablets, laptops, desktops and an ongoing increase in the kind of devices and form factors.
* The speed of development is one of its major advantages. If you want to develop an application or a website promptly, it is imperative to consider using Bootstrap. It helps to save your coding effort by offering less CSS functionality and pre-built blocks of code rather than structuring code from the scratch.
* Ready-made themes of Bootstrap will help achieve your needs through a faster route. It facilitates abundant customization and helps developers in designing tailor made websites, according to their specifications. It has the facility to select any feature that is actually needed to create a customized website.
  1. **Role and Profile**

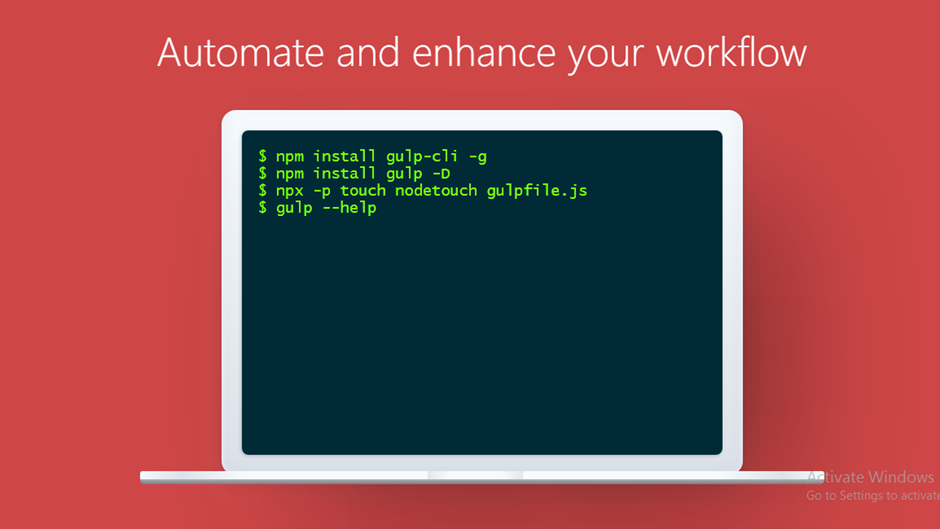
I was responsible for the coding, design and layout of a website according to a company’s specifications. As the role takes into consideration user experience and function, a certain level of both graphic design and computer programming is necessary. Once a website has been created, a Web Developer will generally assist with the maintenance and upkeep of the website. A Web Developer must have experience in the planning and the delivery of web applications across multiple platforms. A typical Web Developer job description should cover:

1. Writing efficient code.
2. Creating websites/a website using standard HTML/CSS/BOOTSTRAP
3. Working closely with web designers and programmers to produce the website.
4. Constant communication with other colleagues in the business to develop and deploy their content- and ensuring there is a clear establishment of what can be created with in what timeframe.
5. Researching different software programs, maintaining software documentation. Implementing contingency plans in case the website goes down.
6. Maintaining and expanding/enhancing the website once built.
7. Managing a team might also be part of job role.

**CHAPTER 2: GULP.JS**

**2.1Objectives of the work undertaken**

* Task-runners like gulp built on Node.js rather than [npm](https://en.wikipedia.org/wiki/Node_Package_Manager), because the basic npm scripts are inefficient when executing multiple tasks. Even though some developers prefer [npm](https://en.wikipedia.org/wiki/Node_Package_Manager) scripts because they can be simple and easy to implement, there are numerous ways where gulp seem to have an advantage over each other and the default provided scripts.
* Gulp runs tasks by transforming files and saves as new ones in temporary folders and the output of one task is taken as input for another and so on until the output reaches the destination folder. This involves a lot of [I/O](https://en.wikipedia.org/wiki/I/O) calls and creation of many temporary files. Whereas gulp streams through the file system and does not require any of these temporary locations decreasing the number of I/O calls thus, improving performance.
* gulp is a build tool in JavaScript built on [node streams](https://en.wikipedia.org/wiki/Node_stream). These streams facilitate the connection of file operations through [pipelines](https://en.wikipedia.org/wiki/Pipeline_(software)). gulp reads the file system and pipes the data at hand from one single-purposed plugin to another through the .pipe() operator, doing one task at a time.
* The users can also write their own plugins to define their own tasks. Unlike other task runners that run tasks by configuration, gulp requires knowledge of JavaScript and coding to define its tasks. gulp is a build system which means apart from running tasks, it is also capable of copying files from one location to another, [compiling](https://en.wikipedia.org/wiki/Compiling), [deploying](https://en.wikipedia.org/wiki/Software_deployment), creating notifications, unit testing, linting, etc. The gulp tasks are run from a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) shell and require two files, package.json, which is used to list the various plugins for gulp, and gulpfile.js.

****

**Figure 2.1: Installation of gulp.js**

**2.1Scope of the Work**

Gulp is a build system for automating tasks. The following is few listing which can be achieved with the help of gulpjs. For any changes and then perform some kind actions on it for eg. Assume you change the styles.css or script.js files which are used for development purpose and later you want to push it for production then you need to minify it manually, instead of you minifying it gulp does it for you. In my project I used gulp.js to manage files or assets.

* Move files to another location (ex from project folders to Web folder)
* Combine files (concatenate) into another file
* Convert files to another type (ex: less to css)
* Minimize files (making it small)
* Make it do it every time a file is modified
* Make certain modifications only on certain conditions (Minimizing files in dev environment is a waste of time, do it in prod)

These are not the only uses it can have but that are why I used it for.

**VARIOUS PLUGINs IN GULP.JS**

**1. gulp-sass:**

Gulp-sass is one of the famous plugin or package that used for converting the scss file to css file automatically using gulp.

**Steps to use this plugin-**

**Step-1:** First you need to install this package. We need use this command in node.js command prompt-

**npm i gulp-sass**

**Step-2:** To execute this plugin we need to have 1 or multiple scss file

**Step-3:** Code to execute-

var scss\_css = function(done) {

  gulp

    .src("src/styles/\*.scss")

    .pipe(sass())

    .pipe(gulp.dest("temp/styles"));

    done();

};

**2. gulp-clean-css:**

gulp-clean-css is one of the common plugin or package that used for minify the css files automatically using gulp by writing few lines of code.

**Steps to use this plugin-**

**Step-1:** Install the package or plugin using this command on node.js command prompt-

**npm i gulp-clean-css**

**Step-2:** To execute this plugin we need to have at least one css file.

**Step-3:** Code to execute-

var minify\_css = function(done) {

  files = ["src/styles/\*.css"];

  fs.exists(files, function(exists) {

    gulp

      .src(files)

      .pipe(CleanCSS())

      .pipe(gulp.dest("temp/styles"));

    done();

  });

};

**3. gulp-concat-css:**

gulp-concat-css plugin is used for concate the different css file into one main css file. Like I have one p.css, q.css, r.css, s.css and we want to concate in one .css file.

**Steps to use this plugin-**

**Step-1:** Install the package or plugin using this command on node.js command prompt-

**npm i gulp-concat-css**

**Step-2:** To execute this plugin we need to have 1 or multiple css file.

**Step-3:** Code to execute-

var concate\_css = function(done) {

  return gulp

    .src("src/styles/\*.css")

    .pipe(concat("src/styles/concat/concate.css"))

    .pipe(gulp.dest("temp/styles/\*.css"));

};

**4. gulp.spritesmith:**

gulp.spritesmith is one of the famous and important plugin for images. This plugin help us to collect all the images and make it spritesheet. In this spritesheet, there are multiple images combine it into one and give it us a new name to image. By making the spritesheet, it also make style sheet as well where it provides all properties to image that they need to make perfect spritesheet.

It also helps to increase the performance of website also.

**Steps to use this plugin-**

**Step-1:** Install the package or plugin using this command on node.js command prompt-

**npm i gulp.spritesmith**

**Step-2:** To execute this plugin we need to have 1 or multiple css file.

**Step-3**: Need to provide the image name and css name that create by executing.

**Step-4:** Code to execute-

function image\_sprite(done) {

  var spriteData = gulp.src("src/image/\*.\*").pipe(

    spritesmith({

      imgName: "image\_sprite.png",

      cssName: "image\_sprite.css"

    })

  );

  spriteData.img.pipe(gulp.dest("temp/images/"));

  spriteData.css.pipe(gulp.dest("temp/styles/\*.\*"))

  done();

}

**5. gulp-imagemin:**

gulp-imagemin plugin is used to optimize the image by removing the extra space in images. It also help to increase the performance of website.

**Steps to use this plugin-**

**Step-1:** Install the package or plugin using this command on node.js command prompt-

**npm i gulp-imagemin**

**Step-2:** To execute this plugin we need to have at least one image.

**Step-3:** Code to execute-

function minify\_image(done) {

  files = ["src/image/\*.\*"];

  fs.exists(files, function(exists) {

    gulp

    .src(files)

    .pipe(imagemin())

    .pipe(gulp.dest("temp/image"));

    done();

  });

}

**6. gulp-uglify:**

gulp-uglify package is used to minify the javascript file. It increases the website performance as well.

**Steps to use this plugin-**

**Step-1:** Install the package or plugin using this command on node.js command prompt-

**npm i gulp-uglify**

**Step-2:** To execute this plugin we need to have at least one javascript file.

**Step-3:** Code to execute-

function js\_minify(done) {

  files = ["src/js/\*.\*"];

  fs.exists(files, function(exists) {

    gulp

      .src(files)

      .pipe(uglify())

      .pipe(gulp.dest("temp/js/"));

    done();

  });

}

**2.2 Importance and Applicability**

### The number of files in a project grows daily, and so do the places these files get imported in. Under an ES5/Angular 1.x environment normally all files are included inside an index.html file, however when working with a more complex environment which includes testing, coverage reports, and automated builds, it pays off to automate the imports, and even try to unify them into a few generic importing tasks that can be reused in multiple workflows.

### Gulp has quite a few great plugins for dealing with such automated imports, [gulp-inject](https://www.npmjs.com/package/gulp-inject) being a few good examples; however initial configuration could be challenging.

### 2.2.1 Understanding the directory structure

A well thought out directory structure can go a long way in preventing issues that can occur in the era of precompiled languages, transpilers and multiple environment modes. **Due to this we understand that:**

* Our source code is not the same code (and most of the time not in the same programming language) that's being served to the browser.
* Our source code is served minified and concatenated in production, but this can be confusing for development.

**2.2.2 Provide distinct development and production builds-**

With the right directory structure, injection automation and build automation in place, we can use browser-sync, the tool recommended by the gulp team, to serve either a production or a development version of our app.

**2.2.3 Inject files with gulp-inject-**

Between [gulp-inject](https://www.npmjs.com/package/gulp-inject) there is no excuse to ever have to manually inject a file into index.html. gulp-inject does a great job at selecting all your java script and css files.

**2.2.4 Separate Gulp tasks into multiple files-**

As the project grows, so does the number of gulp tasks. This can leave you with a very large and messy gulpfile.js. A usefull trick is to split the file into multiple files inside a directory, then use require-dir to require all the task files within gulpfile.js.

var requireDir = require("require-dir");

requireDir("./gulp-task");

**2.3 Role and Profile**

* There are several key features regarding the use of Gulp which would make you want to use it. The one I hold as the most important is the way it can **simulate** the server environment where you will ultimately be hosting your code. This includes moving files around your project directory, and more importantly placing them in a **development directory** where you will be running a web server.
* Gulp also enables you to **compile**, **minify** and **concatenate** any files you want. All with the sole goal of getting your code base’s footprint down to the absolute minimum. In this process making it ready for **shipping to production**. It’s perfectly fine if you don’t know any of the terms above, we’ll go through them in more detail a bit further down.

**CHAPTER 3: KOA.JS and MONGO DB**

* 1. **Objectives of the work undertaken**

Koa is a new web framework designed by the team behind Express, which aims to be a smaller, more expressive, and more robust foundation for web applications and APIs. By leveraging async functions, Koa allows you to ditch callbacks and greatly increase error-handling. Koa does not bundle any middleware within its core, and it provides an elegant suite of methods that make writing servers fast and enjoyable.

MongoDB describe themselves as a general purpose database but IMO the sweet spot for the DB is in the development of modern applications with rapidly changing requirements utilizing modern programming stacks.

MongoDB has the following strong advantages when building a modern application:

* The flexible schema allows you to iterate through product designs faster than databases like Oracle, since changes to the data model don’t require database reconfiguration or rebuild
* Applications build with Mongo DB can easily integrate with Continuous Integration platforms, since all the data model is encapsulated within the program code.
* MongoDB reduces the friction between object model and database model - there’s no need to always convert between the representation of data in the application and that in the database. That having been said, carefully constructing the data model is still essential for optimal performance.
* The use of JavaScript as the database language is very convenient when programming in JavaScript elsewhere in the application - particularly if you are using NodeJS
* MongoDB is open source and so can be used without licensing costs (though MongoDB the company offers enterprise and hosted options that have significant advantages over the open source version).
* MongoDB is relatively easy to setup up, and scaling is relatively straight forward. Administration of MongoDB in production is less expensive than for many other databases.
  1. **Scope of the Work**

**SARA** isa chatbotthat is on development phase. It is developing in a way that it is helpful for end customers interactions to answer their questions. Our Team is trying to make it more interactive so that the end customers will benefit from it. Its services are provided from eClerx client website.

**TECHNOLOGIES USED TO DEVELOP THIS PRODUCT:**

|  |  |
| --- | --- |
| CATEGORY | TECHNOLOGIES |
| WEB DEVELOPMENT | Angular 8, Nodejs, Koa server |
| NO SQL DATABASES | MongoDB |

**SOFTWARES REQUIRED TO INSTALL:**

* Angular 8 CLI
* Node runtime environment
* MongoDB
* Robo 3t

**3.2.1 Angular CLI:**

This Framework is used to design the web layout for the project, Typescript is the preferred language in the Angular 8.

**Advantages are mentioned below-**

* Component-based architecture that provides a higher quality of code
* Google Long-Term Support
* Seamless updates using Angular CLI
* High Performance
* Loved by millions of developers
* Unit-test friendly

**Use of this software in project:**

* By using this framework, our team create the interface of the chatbot.
* Routing played the key role in navigating and rendering the components using router outlets.
* Sharable data is accessible through user data services.
* Created ts (type script) files for every component to implement the functionality.
* Imported plugins for making chats and api connections
* Create the admin portal for the chatbot, through which admin can see all the chatlogs of all the users.

**Admin portal is divided into four sections**

* 1. Landing home page
  2. Dashboard
  3. Chatlogs
  4. Statistics
  5. Priority messages
  6. Profile page

**3.2.2 Node Runtime environment:**

* Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.
* nodejs provide the non-blocking I/O.
* Initially javascript is restricted to client-side programming only which is run by the browser only but nodejs changes everything about the javascript giving javascript power of programming both for client and server side.
* Node js supports npm packages which attracts a greater number of developers to develop packages for the nodejs.
  + - 1. **Use of nodejs and Koa in Project:**
* By using nodejs we made the api for chatbot appilication
* List of api’s
* **Login** – which made available by running the nodejs and koa server and accepts get and posts requests from the angular.
* **Register** – Which is used to register the user, The user data is taken from the register form which is made by using angular, and the data is post to the api url .
* **Chatlogs** – This API returns the chatlogs of user.

**3.3Importance and Applicability**

**HTTP Methods:** Koa.js identifies and understands the operation required by a client through HTTP methods like GET, POST, PUT and DELETE. Hence, the developers need to ensure that each request sent from the client includes the appropriate HTTP method. For instance, a request to retrieve data must contain the GET method, whereas a request sent to the server to accept enclosed data must include the POST method.

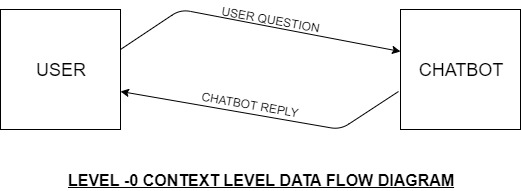
**3.3.1 Register API-**

**Register API is used to authenticate the user. When the client went to register form then they need to fill the form otherwise it shows the validation error.**

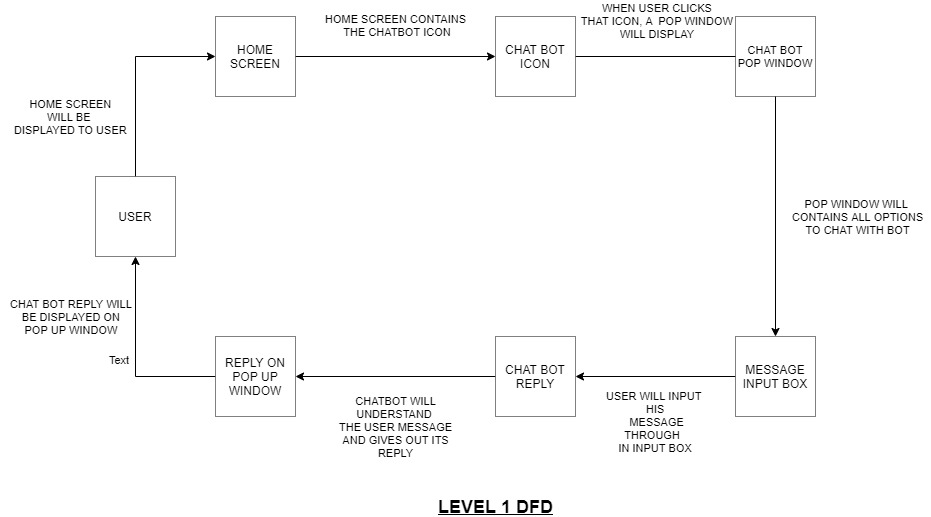
**In Node.js, there are some plugins that we can used to develop the API’s.**

**3.3.1 DFD’s**

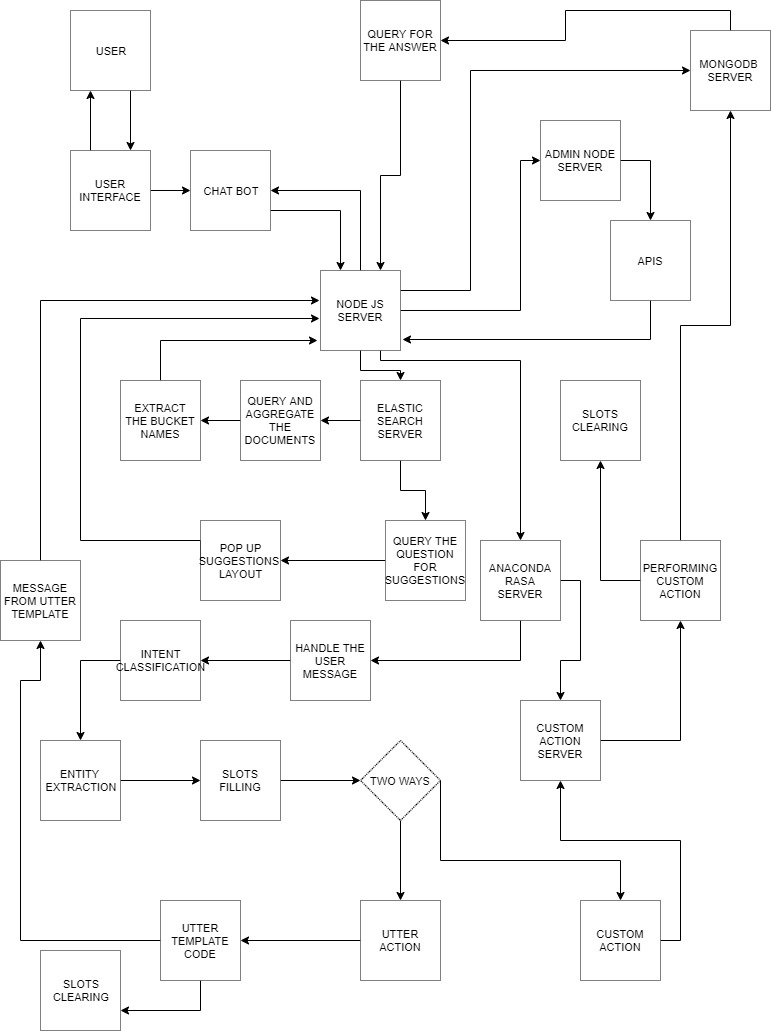
**3.3.1.1 Level 0 Diagram-**

****

**3.3.1.2 Level 1 Diagram-**

****

**3.3.1.3 Level 2 Diagram**

****

**3.3.1.2 Plugins to develop the register API-**

**1. koa-**

**Koa is one of the main plugin which helps to create the server as same as apache.**

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i koa**

**Step-2: After Install the plugin then we need to require on main that generally execute.**

const koa = require("koa");

**2. koa-router-**

koa-router plugin is used to route the API’s. It express the style routing using app.get, app.put, app.delete, app.post.

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i koa-router**

**Step-2: After Install the plugin then we need to require on main that generally execute.**

const koaRouter = require("koa-router");

const router = new koaRouter();

**3. koa-body-**

Koa-body plugin is used to read the HTTP post data.

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i koa-body**

**Step-2: After Install the plugin then we need to require on main that generally execute.**

const bodyParser = require("koa-body");

**4. bcryptjs-**

Bcryptjs is the one of the important plugin that is used to hash the password. It means that it encoded the password.

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i bcryptjs**

**Step-2: After Install the plugin then we need to require on main that generally execute.**

const bcrypt = require("bcryptjs");

var hashedPassword = bcrypt.hashSync(ctx.request.body.password, 10);

**5. datalize-**

Datalize is the plugin which used to validate the forms. This can be used in express as well as koajs.

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i datalize**

**Step-2: After Install the plugin then we need to require on main that generally execute**

const datalize = require("datalize");

const field = datalize.field;

**MONGODB PLUGINS-**

**1.mongoose-**

* Mongoose is a **MongoDB** object modeling tool designed to work in an asynchronous environment. Mongoose supports both promises and callbacks.
* Mongoose provide the schema property and model as well.

**Steps to use this plugin-**

**Step-1: We need to install this plugin. To install this plugin we need to write the command on node.js command prompt-**

**npm i mongoose**

**Step-2: After Install the plugin then we need to require or write the code as well-**

const mongoose = require("mongoose");

const assert = require("assert");

mongoose.connect(

  process.env.DB\_URL,

  {

    useNewUrlParser: true,

    useUnifiedTopology: true,

    useCreateIndex: true

  },

  function(error, link) {

    assert.equal(error, null, "DB CONNECT FAIL");

    console.log("DB CONNECT");

    console.log(link);

  }

);

**Step-3: We need to make the schema and model using mongoose plugin**

var mongoose = require("mongoose");

var UserSchema = mongoose.Schema({

  name:{

   type: String,

   required:true,

  },

  email: {

    type: String,

    required: true,

    unique: true

  },

  password: {

    type: String,

    required: true

  },

  isActive: {

    type: Boolean,

    default: true

  },

  CreateOn: {

    type: Date,

    default: Date.now()

  }

});

//model

const models = mongoose.model("users", UserSchema);

module.exports = models;

**FUNCTIONS USED TO DEAL WITH DATABASE**

**1.db.collecton.save() Method-**

Save method is used to insert the client information in database.

**Code to Run-**

const temp = new User();

    temp.name = ctx.request.body.name;

    temp.email = ctx.request.body.email;

    temp.password = hashedPassword;

    await temp

      .save()

      .then(data => {

        ctx.body = data;

      })

      .catch(err => {

        ctx.body = "error: " + err;

      });

**2.db.collection.findOne() Method-**

Find method is used to read the data from database and check whether user exist or not.

**Code to Run-**

await User.findOne({ email: ctx.request.body.email }, function(

      error,

      result

    ) {

      if (error) {

        return (ctx.body = {

          status: false,

          message: "User does not exist"

        });

      }

      if (result) {

        const isMatch = bcrypt.compareSync(

          ctx.request.body.password,

          result.password

        );

        if (isMatch) {

          ctx.body = {

            status: true,

            message: "User exist"

          };

**Brief description of the work done**

**1. Position of Internship and roles**

Compile and analyze data, processes, and codes to troubleshoot problems and identify areas for improvement. Collaborating with the front-end developers and other team members to establish objectives and design more functional, cohesive codes to enhance the user experience. Developing ideas for new programs, products, or features by monitoring industry developments and trends. Recording data and reporting it to proper parties, such as clients or leadership. Participating in continuing education and training to remain current on best practices, learn new programming languages, and better assist other team members. Taking lead on projects, as needed.

**2.Activities/ equipment handled**

**MongoDB:** MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that we need.

**Gulp.JS:** Gulp is a toolkit for automating painful or time-consuming tasks in your development workflow, so you can stop messing around and build something.

**Node.JS:** Node.js is a platform built on [Chrome's JavaScript runtime](https://code.google.com/p/v8/) for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

**Koa.JS:** Koa.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the creators of Express.js, the most popular node web framework.

**Bootstrap:** Bootstrap is an open-source Javascript framework developed by the team at Twitter. It is a combination of HTML, CSS, and Javascript code designed to help build user interface components. Bootstrap was also programmed to support both HTML5 and CSS3.

**3. Challenges faced and how those were tackled**

**3.1. Scalability**

Scalability is a difficult thing for web developers to manage. Scalability is load balancing between the servers, hence when the load increases (i.e. more traffic on the page) additional servers can be added to balance it. All the load should not be thrown on a single server instead of which the software should be designed in such a way that it can work on multiple servers. Service oriented architecture helps developers in managing and improving scalability.

**3.2. UI/UX**

In the era of smartphones, web developers are expected to develop UI/UX that is responsive and user-friendly. If the web applications frustrate users, then it is difficult to maintain the customer’s loyalty on your website. Website navigation is another part often neglected by developers. Intuitive navigation creates a better user experience for the website visitor.

**3.3 Performance**

Slow web applications are a failure and as a result, customers abscond your website, thus damaging your revenue as well as reputation. Some of the performance issues developers’ faces are Poorly written code, Un-Optimized Databases, Unmanaged Growth of data, Traffic spikes, Poor load distribution, Default configuration, Troublesome third party services, etc.

**3.4. Knowledge of Framework & Platforms**

Frameworks are boost performance, offer libraries of coding and extend capabilities, so developers need not do hand-coding web applications from the ground up. Frameworks offer features like models, APIs, snippets of code and other elements to develop dynamic web applications.

**3.5. Security**

Security is something the web developers need to consider at every stage of SDLC (software development life cycle). There are many things to consider when it comes to web application security such as denial of service attacks, the safety of user data, database malfunctioning, unauthorized access to restricted parts of the website, etc.

**Learning outcomes**

The backend usually consists of three parts: a server, an application, and a database. If you book a flight or buy concert tickets, you usually open a website and interact with the frontend. Once you’ve entered that information, the application stores it in a database that was created on a server. For sake of ease, just think about a database as a giant Excel spreadsheet on your computer, but your computer (server) is stored somewhere in Arizona. All of that information stays on the server so when you log back into the application to print your tickets, all of the information is still there in your account.

**CONCLUSION**

* The Bootstrap framework provides an easy way of crafting website designs and allows creating greatly optimized mobile designs. It efficiently scales up the website and applications with a single code base.
* Bootstrap grid system builds mobile first, responsive websites, for small screens and then scales up the designs from there for the devices with larger display screen.
* By preferring code over configuration, node best practices, and a minimal API surface - gulp makes things simple like never before. Using the power of node streams, gulp gives you fast build that don't write intermediary files to disk.
* Node is a JavaScript environment has some great features that make it an attractive choice for building server-side application middle tiers, including web servers and web services for platform APIs.
* MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. The document model maps to the objects in your application code, making data easy.
* Koa.js is a widely used Node.js web application framework. Koa.js is developed and maintained by the creators of another widely used Node.js framework — Express.js

**REFRENCES**

1. www.gulpjs.com
2. [www.google.com](http://www.google.com)
3. [www.koajs.com](http://www.koajs.com)
4. [www.w3schools.com](http://www.w3schools.com)
5. [www.wikipedia.com](http://www.wikipedia.com)
6. [www.bootstrap.com](http://www.bootstrap.com)
7. [www.github.com](http://www.github.com)