*Practical Assignment Report On*

**Employee Appraisal CRUD Application**

for the course

**IT254 : Web Technology and Applications**

*Submitted by*

**Shashikantha (181IT242)**

**IV SEM B.Tech (IT)**

*Under the guidance of*

**Name of Guide**

**Dept of IT, NITK Surathkal**

*in partial fulfillment for the award of the degree*

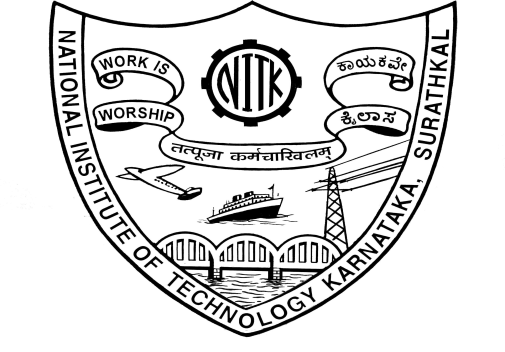
*of*

**Bachelor of Technology**

in

**Information Technology**

at



**Department of Information Technology**

**National Institute of Technology Karnataka, Surathkal**.

***June 2020***

**Abstract**

Employee appraisal form is mainly used for evaluating the performance of the employee belonging to the particular organization. And in recent years this is automated using web applications. And this is very helpful for submitting the appraisal form for employees by the admin. And it helps in efficient evaluation of the employee by monthly or weekly purpose. And modern web architecture mainly uses the modern frontend and backend framework. And these new frameworks reduce a lot of work compared to old web application design. And single page application is mainly used all over the world and this renders the whole component in a single page makes it efficient in rendering the object. And properly handling the back page navigation.

**Tables Of Contents**

**1 . Introduction** --------------------------------------------------- 4

**2 . Literature Survey**  ---------------------------------------------- 4

**3 .System design /architecture**------------------------------------- 4

3.1 Normal MVC workflow ---------------------------------- 4

3.2 Extended architecture explained--------------------------5

**4 Methodology----------------------------------------------------------6**

**5 Implementation-------------------------------------------------------6**

5.1 Auth Module -------------------------------------------7

5.2 Employee Module -------------------------------------9

5.3 Admin Module-----------------------------------------10

**4 .Conclusion------------------------**------------------------------------11

**1. Introduction**

Organizations are run and steered by people and their labors. It is through people that goals are set and objectives realized depend on the performance of the employees.The performance of an organization is thus dependent upon the sum total of the performance of its members. The success of an organization depends on its ability to measure accurately the performance of its members and use it objectively to optimize them as a vital resource and strength.This website is very helpful for evaluating the performance of the employee based on monthly or weekly basis.The performance of an employee is his resultant behavior on task, which can be observed and evaluated.It refers to the contribution made by an individual in the accomplishment of organizational objectives.Performance can be measured by some combination of quantity, quality, time, and cost.This project is designed using MVC architecture angular 9 is used as a frontend framework and node is used as backend and mongoose is used as a database for storing the data. This application is secured using jsonwebtoken only registered members can access the database, execute the database query and can get the record and all the data is stored as a document in the database. I have tried to make use of the best practices while designing this application

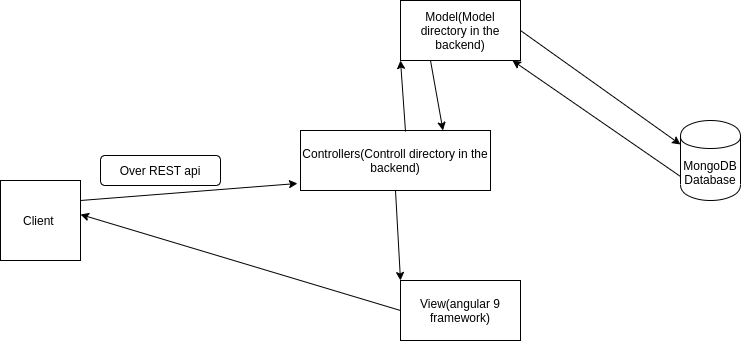
**2 .Literature Survey**

Although the interest in and use of performance appraisal have increased over the past 30 years, the practice of formally evaluating employees has existed for centuries. The main motivation for using the MVC architecture in this project is to handle the multiple schema and multiple functionality. And because of this MVC architecture we can use this powerful tool to completely separate each and every part of the application and make it more universal. Which helps in easy handling of the application for different platforms. Another motivation is the angular material library from google which helps in easy and flexible interface design which attracts the user. And using material in “angular” web application makes it very easy to handle the frontend component and saves a lot of time.

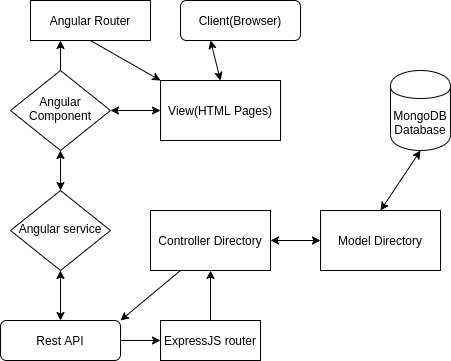
**3.System Design/Architecture**

**3.1.Normal MVC Workflow**

This project follows the standard MVC workflow where Angular works as a view and preprocess the html for getting and sending data over rest api which is connected to node js backend. Node js back end contains the main two directory model and controller all the requests are redirected to controller controller which in turn send the request to model which is model directory in our case and model directory communicates with backend and fetch the raw data from database these data is preprocessed and converted in the requested format by the controller when the model sends back the request to the controller.Controller inturn send the response back to the requested endpoint over REST api. Data is preprocessed in the angular framework in appropriate format by angular component and displayed to the client .



**3.2 Extended architecture explained**

This section explains each and every part of the application using standard MVC architecture in more detail. The front end framework angular is used to manage the incoming request and send the request. The first part that is invoked when a page is loaded is angular router which makes appropriate route calculation and redirects to the requested destination.When that destination page is loaded the actual angular component will fetch the data by sending the request to backend and retrieves the data. This is done using special type of files known as service files which makes the request to backend ones the data is fetched it is populated in the front end.When backend receives the request express js router will take care of the routing and redirects to the requested backend route it will call the function defined on that specific route. These functions are defined in the controllers directory and these controllers make the request to model to retrieve the appropriate data and model retrieve the data from backend and send that back to controller and controller send the response to the request previously sent.

**4 Methodology**

The whole front end is developed using angular 9 framework and whole backend designed using the node js backend As explained above MVC architecture is used for design the whole web site.My project folder consist of two main folder client and server client will handle whole client side application and server will handle the whole backend server directory contains three main directory Routers, Models, Controllers all of these directory handle the request in integrated approach. Controllers directory has the Auth controller directory which is used for handling the authentication. Client directory consists of three modules employee and admin and auth module where admin modules have all the components related to handle all the frontend logic of admin and employee module will have the components for handling the frontend logic of employee. Auth module is used for handling the authentication .In order to communicate with the backend I have used the mongoose package. Which provides a wide variety of functionality for handling the database.

**5 Implementation**

**5.1 .Auth Module**

Auth folder contains three component “registeruser” and “registeradmin” and “login” module.”registeruser” module is used for registering the user with name, organization id, email etc..

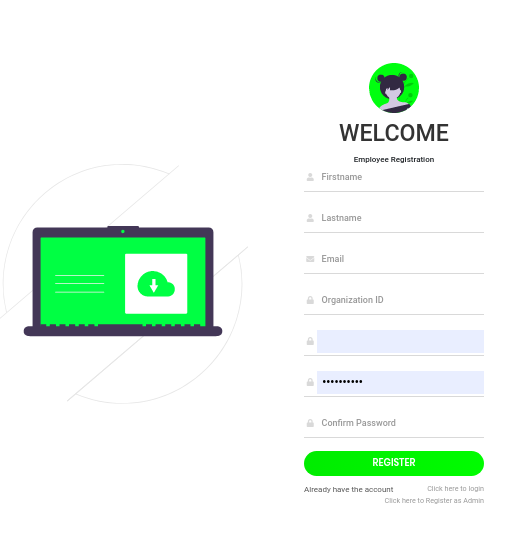


Image 1 : User registration

And registeradmin component will be used for registering the admin for specific organization with organization id and other attribute.

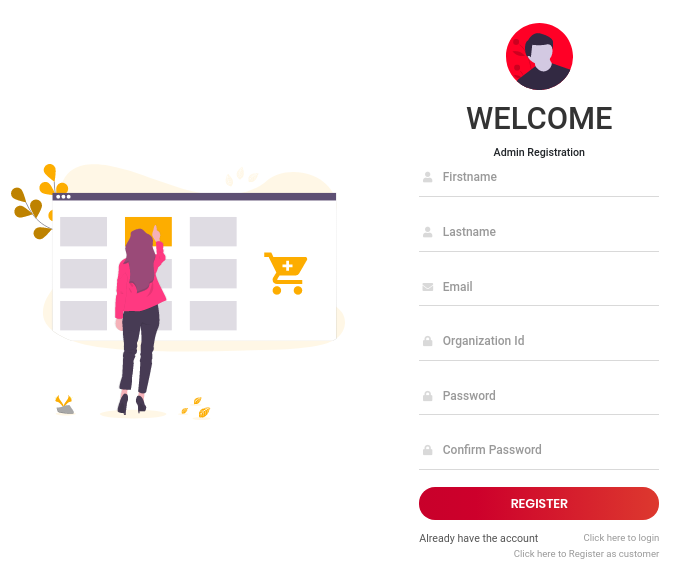
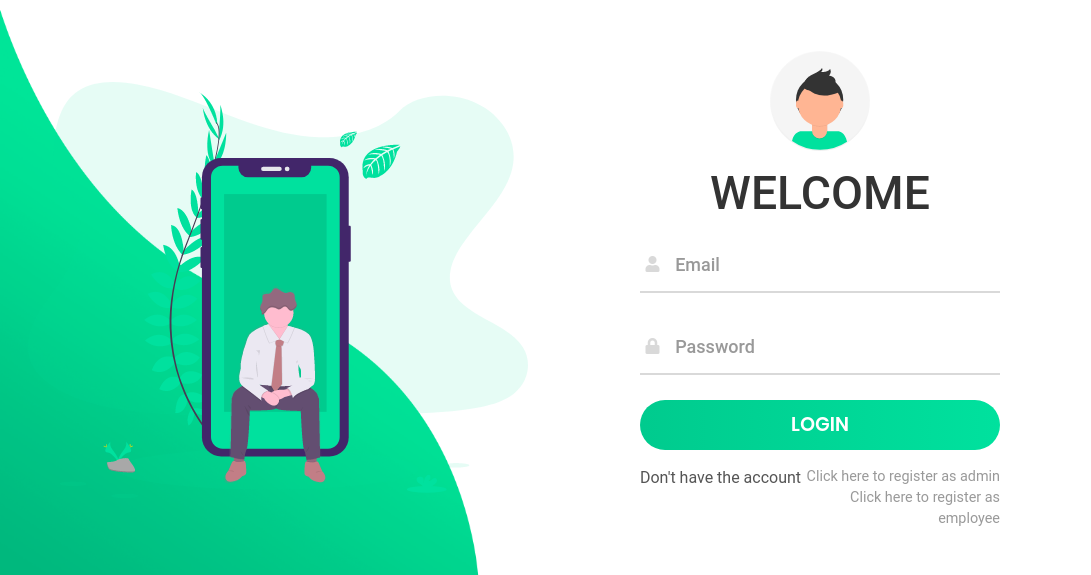


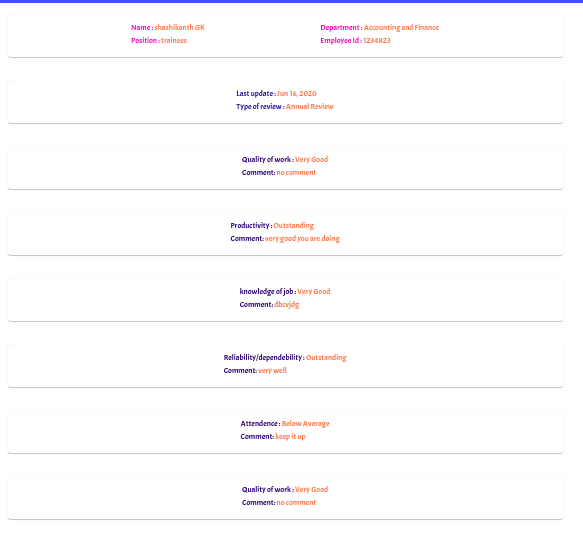
Image 2: Page for registering the Admin for organization

And the third component is the login component and this login component will be used to log in to the account after that json web token from the backend will be stored in the localstorage and used for authentication later.



**5.2.Employee Module**

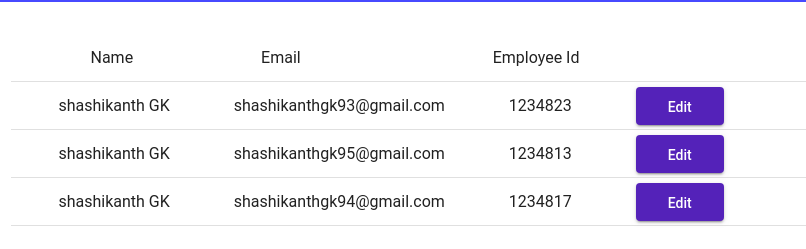
In the Employee module we have a service directory that directly communicates with the backend for retrieving employee appraisal form for a particular employee. Once the data is loaded it will be populated in the html view and the specific employee can retrieve only his record and not others because the backend route is protected. And it also has the navbar component which is used for navigating.



Page 4: Employee appraisal from

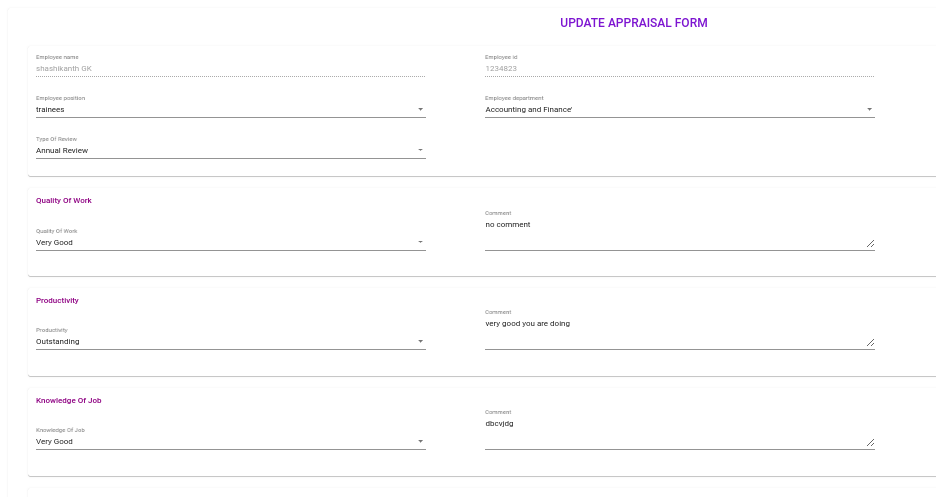
**5.3. Admin Module**

Admin module has three component “updateemployee” , ”submitemployee”, ”getemployee” and admin service directory. Admin service directory is the main directory which contains the service file which is used for handling the request to the backend server. “Getemployee” component is used for getting the employee for currently logged in admin and he can not get the employee name of the other admin as the backend route is protected. Once it gets the data from the database it will be populated in an html file. And it has a list of employees with id and email and name and an option for editing the appraisal form and this appraisal form update component will be rendered if the employee has the previous appraisal form otherwise the submit appraisal form will be loaded.



Page 5: page for getting the employee in organization.

Update employee and submitemployee component have the similar html part only difference is in update employee component all the previous data will be filled in the respective field and submitemployee will have the blank field for filling the data update employee also have the option for deleting the data from the backend and that record is dropped from backend permanently. All the fields are validated before it is submitted in both frontend and backend.



Page 6: Page for submitting the appraisal form

That's all about the frontend and backend is written in node js and it supports all basic authentication.

**6. Conclusion**

This is the very basic MVC architecture and it is scalable as all queries are done inside the transaction but it can be improved very much with extended features.

CodeLink:<https://github.com/shashikanthgk/Shashikanth-WTA-assignment>