COURSE: WEB APP DEVELOPMENT

WEB DEVELOPMENT ASSIGNMENT

PURBOSHI DAS - RA1811028010036

NANDINI CHOURASIA - RA1811028010037

POOJITHA YATA - RA1811028010072

SPRING BOOT

Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

We take an opinionated view of the Spring platform and third-party libraries, so that you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

You can use Spring Boot to create Java applications that can be started by using java -jar or more traditional war deployments. We also provide a command line tool that runs “spring scripts”.

Our primary goals are:

- Provide a radically faster and widely accessible getting-started experience for all Spring development.
- Be opinionated out of the box but get out of the way quickly as requirements start to diverge from the defaults.

- Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).
- Absolutely no code generation and no requirement for XML configuration.

ADVANTAGES :

- It is very easy to develop Spring Based applications with Java or Groovy.
- It reduces lots of development time and increases productivity.
- It avoids writing lots of boilerplate Code, Annotations and XML Configuration.
- It is very easy to integrate Spring Boot Application with its Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security etc.
- It follows "Opinionated Defaults Configuration" Approach to reduce Developer effort
- It provides Embedded HTTP servers like Tomcat, Jetty etc. to develop and test our web applications very easily.
- It provides the CLI (Command Line Interface) tool to develop and test Spring Boot(Java or Groovy) Applications from command prompt very easily and quickly.
- It provides lots of plugins to develop and test Spring Boot Applications very easily using Build Tools like Maven and Gradle
- It provides lots of plugins to work with embedded and in-memory Databases very easily.

HOW DOES THIS WORK?

Spring Boot automatically configures your application based on the dependencies you have added to the project by using @EnableAutoConfiguration annotation. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains @SpringBootApplication annotation and the main method.

Spring Boot automatically scans all the components included in the project by using @ComponentScan annotation.

AUTO CONFIGURATION :

Spring Boot Auto Configuration automatically configures your Spring application based on the JAR dependencies you added in the project. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto configures an in-memory database.

For this purpose, you need to add @EnableAutoConfiguration annotation or @SpringBootApplication annotation to your main class file. Then, your Spring Boot application will be automatically configured.

CODE :

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

@EnableAutoConfiguration

public class DemoApplication {

  public static void main(String[] args) {

    SpringApplication.run(DemoApplication.class, args);

  }}
```

MongoDB

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc.

MAIN FEATURES :

Ad-hoc queries

MongoDB supports field, range query, and regular-expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions.

Indexing

Fields in a MongoDB document can be indexed with primary and secondary indices.

Replication

A replica set consists of two or more copies of the data. Each replica-set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

File storage

MongoDB can be used as a file system, called GridFS, with load balancing and data replication features over multiple machines for storing files,This function, called grid file system.

Aggregation

MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single-purpose aggregation methods.

According to MongoDB's documentation, the Aggregation Pipeline provides better performance for most aggregation operations.

Server-side JavaScript execution

JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

ARCHITECTURE :

Programming language accessibility

MongoDB has official drivers for major programming languages and development environments.There are also a large number of unofficial or community-supported drivers for other programming languages and frameworks.

Serverless access

MongoDB Stitchprovides serverless access to MongoDB and other services. Client libraries are available for JavaScript, iOS, and Android.

Management and graphical front-ends

The primary interface to the database has been the mongo shell. Since MongoDB 3.2, MongoDB Compass has been introduced as the native GUI. There are products and third-party projects that offer user interfaces for administration and data viewing.

BUG REPORTS AND CRITICISMS :

Security

Due to the default security configuration of MongoDB, allowing anyone to have full access to the database, data from tens of thousands of MongoDB installations has been stolen.

From MongoDB 3.6, this default behavior was extended to all MongoDB packages across all platforms. As a result, all networked connections to the database will be denied unless explicitly configured by an administrator.

Technical criticisms

In some failure scenarios where an application can access two distinct MongoDB processes, but these processes cannot access each other, This issue was addressed since version 3.4.0 released in November 2016

AngularJS

AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

AngularJS is used as the frontend of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js server runtime environment.

AngularJS's design goals include:

- to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- to decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.
- to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

BOOTSTRAP :

The task performed by the AngularJS bootstrapper occur in three phases after the DOM has been loaded:

1. Creation of a new Injector
2. Compilation of the directives that decorate the DOM
3. Linking of all directives to scope

AngularJS directives allow the developer to specify custom and reusable HTML-like elements and attributes that define data bindings and the behavior of presentation components. Some of the most commonly used directives are:

ng-class

Conditionally apply a class, depending on the value of a boolean expression.

ng-controller

Specifies a JavaScript controller class that evaluates HTML expressions.

ng-init

Called once when the element is initialized.

ng-repeat

Instantiate an element once per item from a collection.

TWO-WAY DATA BINDING :

AngularJS two-way data binding is its most notable feature, largely relieving the server backend of templating responsibilities. Instead, templates are rendered in plain HTML according to data contained in a scope defined in the model. The $scope service in Angular detects changes to the model section and modifies HTML expressions in the view via a controller.

$watch

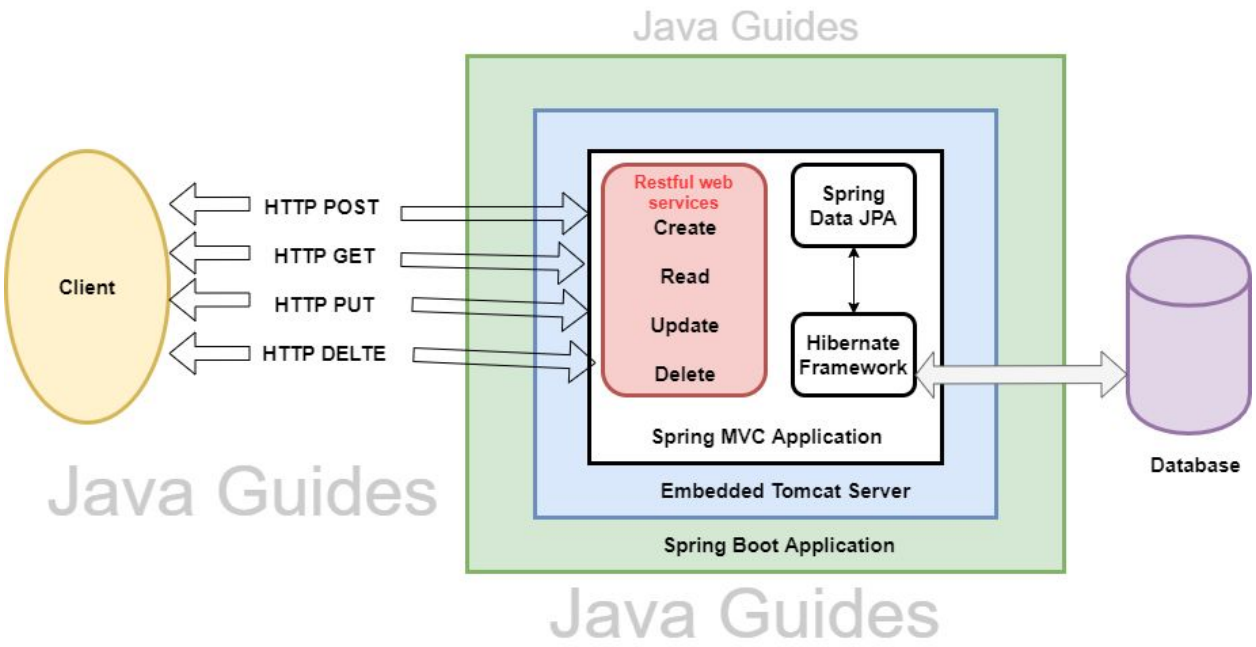
is an angular method used for dirty checking. Any variable or expression assigned in $scope automatically sets up a $watchExpression in angular. So assigning a variable to $scope or using directives like ng-if, ng-show, ng-repeat etc. all create watches in angular scope automatically. Angular maintains a simple array of $$watchers in the $scope objects

Different ways of defining a watcher in AngularJS.

- explicitly $watch an attribute on $scope.
  $scope.$watch('person.username', validateUnique);
- place an interpolation in your template (a watcher will be created for you on the current $scope).
- ask a directive such as ng-model to define the watcher for you.
  <input ng-model="person.username" />

PROCEDURE :

(I)FOR MONGODB:



Spring Boot CRUD Restful API with MongoDB Database

TOOLS & TECHNOLOGIES USED :

1. Spring Boot - 2.1.0.RELEASE
2. Spring Framework - 5.1.2.RELEASE
3. Spring Data - 2.1.2.RELEASE
4. MongoDB - 3.8.2
5. JDK - 1.8 or later
6. Maven - 3.5.1
7. IDE - STS/Eclipse Neon.3

DEVELOPMENT STEPS :

1. Create a Spring Boot Application
2. The pom.xml File - Define Maven Dependencies
   - Run as -> Maven build
   - Run as -> Maven clean
   - Maven ->Update Project
3. Project Structure
4. Configuring MongoDB database
5. Creating the Employee Model
6. Create a Spring Data Repository - EmployeeRepository.java
7. Creating the APIs - EmployeeController
8. Exception handling in RESTFul web services
9. Running Spring boot application[Run as -> SpringBoot Application]
10. Test REST APIs Using Postman Client

(II)FOR ANGULARJS:

TOOLS AND TECHNOLOGIES USED :

Server-side Technologies

1. Spring Boot - 2.0.5.RELEASE
2. JDK - 1.8 or later
3. Spring Framework - 5.0.8 RELEASE
4. Hibernate - 5.2.17.Final
5. Spring Data JPA - 2+

Frontend Technologies

1. Angular 8.0.0
2. Bootstrap 4

3. npm- 6.9.0
4. JQuery

We built a simple Employee Management App (FULL STACK Application with Angular 8 and Spring Boot) with following features:

1. Create an Employee
2. Update an Employee
3. List of Employees
4. Delete Employee
5. View Employee

List of what are components, services, and modules we created in this application.

Components

1. create-employee
2. employee-list
3. employee-details
4. update-employee

Services

1. employee.service.ts - Service for Http Client methods

Modules

1. FormsModule
2. HttpClientModule
3. AppRoutingModule

EXECUTING STEPS :

1. npm install
2. install - g @angular/cli
3. ng serve
4. Localhost:4200

OUTPUT :

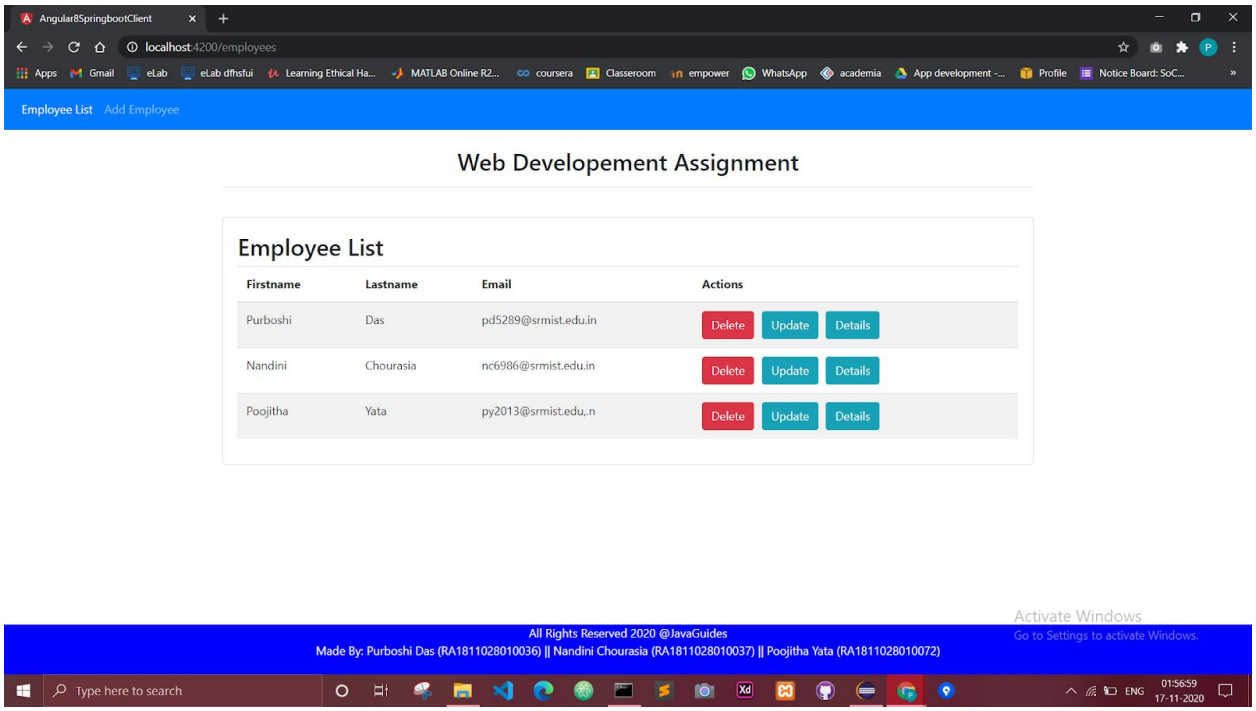


Fig 1.1 Employee List

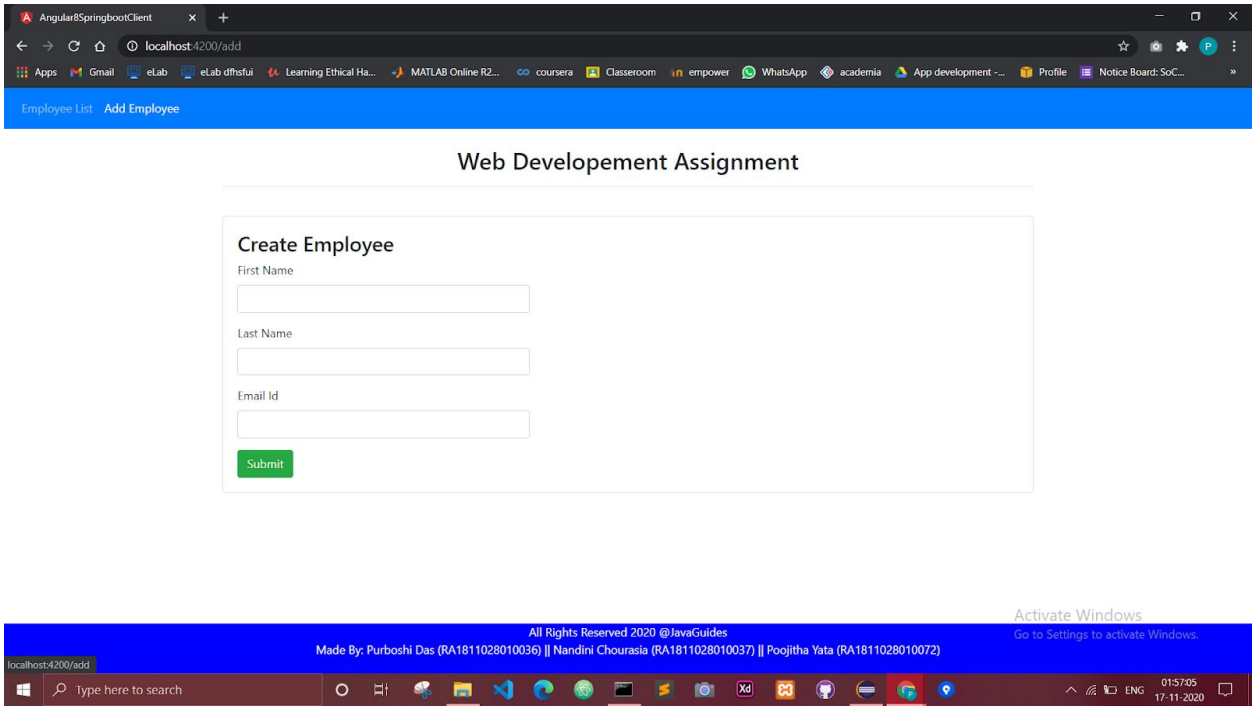


Fig 1.2 Create Employee

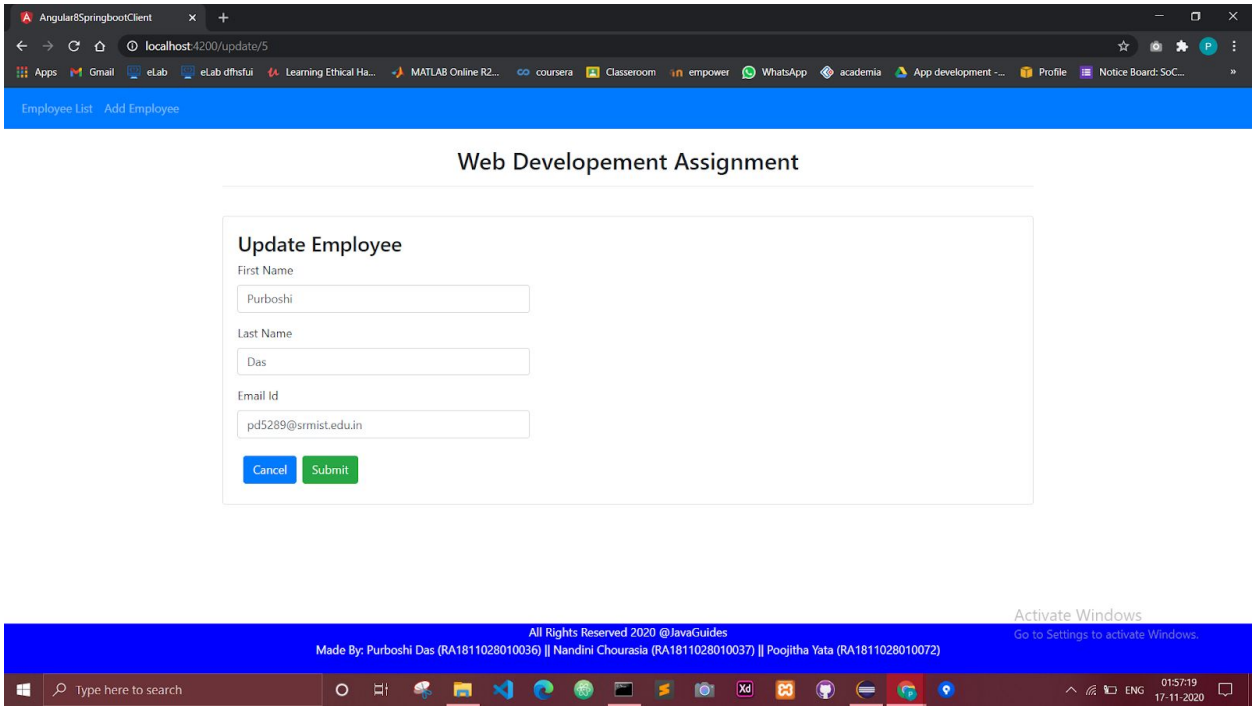


Fig 1.3 Update Employee

CODE FRAMEWORK :

Backend

Fig 1.3 SpringBoot Structure Part 1

Fig 1.4 SpringBoot Structure Part 2

Frontend

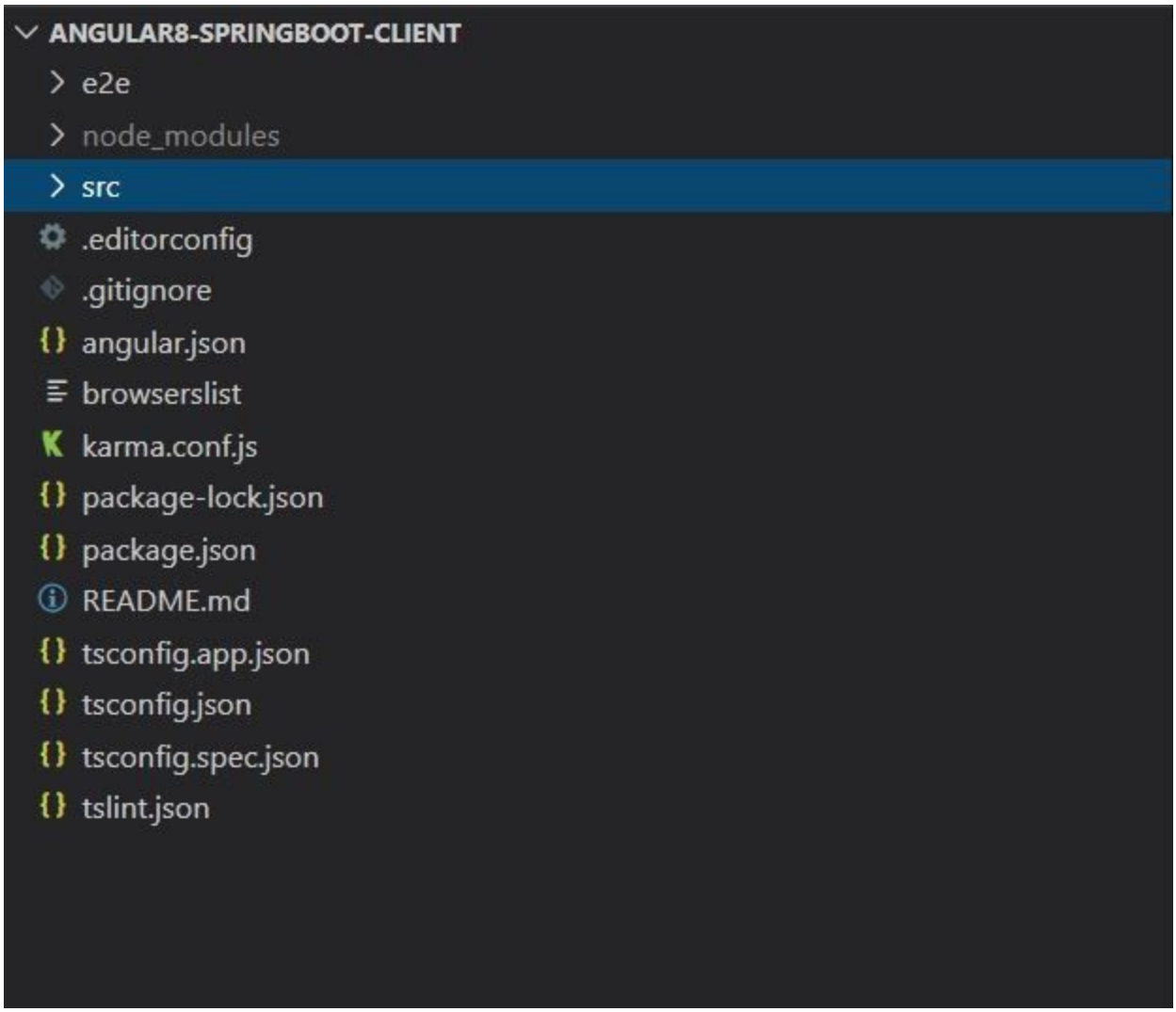


Fig 1.5 Angular Structure Part 1

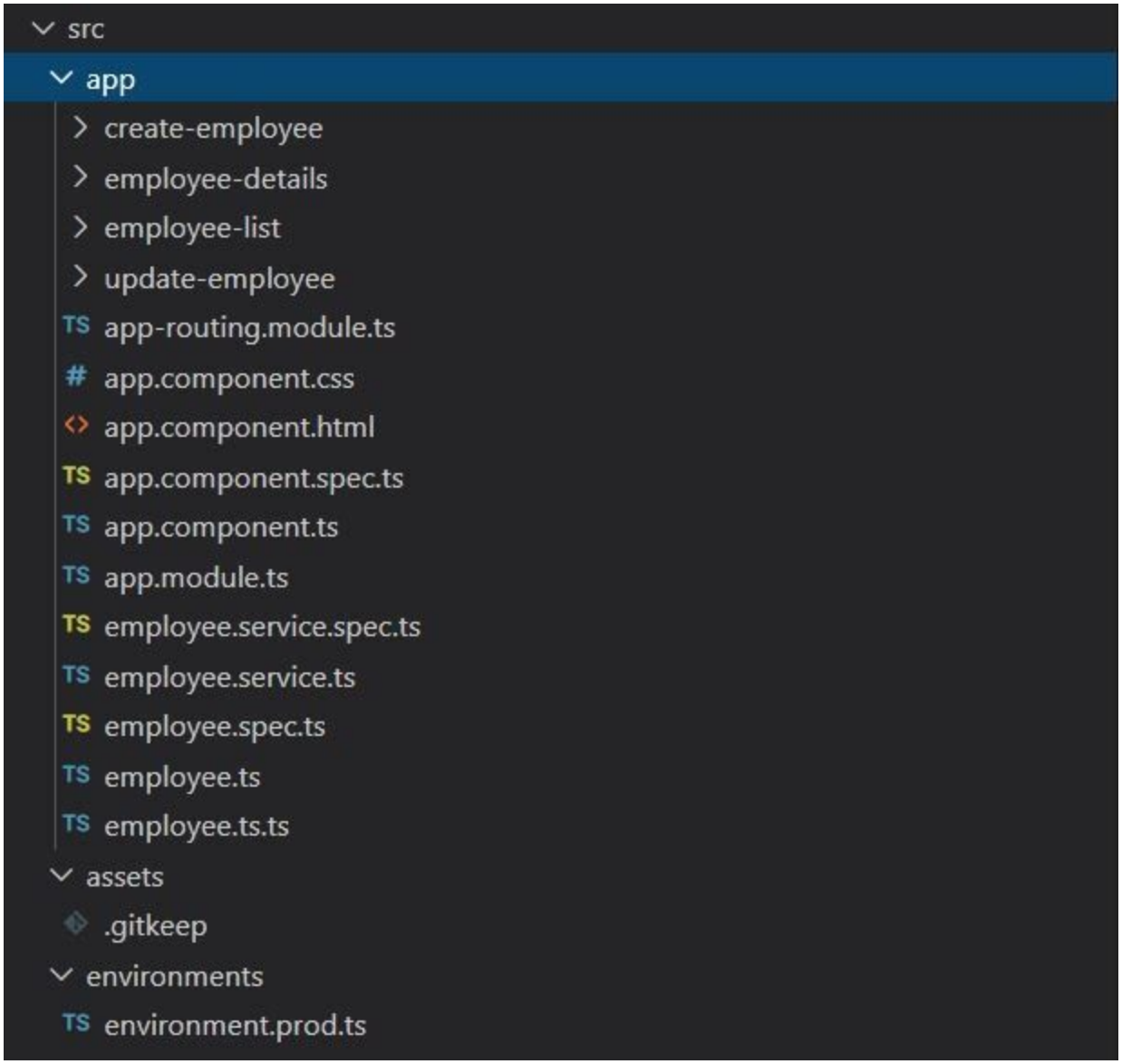


Fig 1.6 Angular Structure Part 2

LINK :

https://github.com/PUR2212/WebDev

MADE BY :-

PURBOSHI DAS - RA1811028010036

NANDINI CHOURASIA - RA1811028010037

POOJITHA YATA - RA1811028010072

Batch - 2
Section - J2