# TRACKIT i- iTRAINING I MANAGEMENT iAPP

*Project ireport isubmitted iin ifulfillment iof imajor iproject iof*

BACHELORS iOF iTECHNOLOGY iIN

Computer iScience iand iEngineering iBy

**VAIBHAV iJINDAL i(171453) PRAKHAR iJAIN i(171475)**

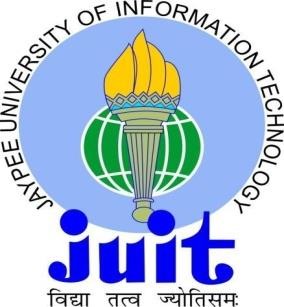
**TARUN iGUPTA i(171382) ii iPRINCE iHARSHVARDHAN i(171370)**

**AMAN iPRAJAPATI i(171459) RADHESH iBHATIA i(171319)**

**SHIKHAR iAGGARWAL i(171300)**

UNDER iTHE iSUPERVISION iOF

**MR. iRISHABH iSHARMA**



DEPARTMENT iOF iCOMPUTER iSCIENCE iAND iENGINEERING iJAYPEE iUNIVERSITY iOF iINFORMATION iTECHNOLOGY iWAKNAGHAT,

SOLAN

# Table iof iContents

[Trackit-Training iManagement iApp i](#_bookmark0)

[Project iReport iUndertaking iv](#_bookmark1)

[ACKNOWLEDGEMENT v](#_bookmark2)

[LIST iOF iFIGURES vi](#_bookmark3)

[ABSTRACT vii](#_bookmark4)

[CHAPTER i1 1](#_bookmark5)

* 1. [Overview 2](#_bookmark6)
  2. [Problem iStatement 3](#_bookmark7)
  3. [Tools i& iTechnologies iUsed 3](#_bookmark8)
  4. [Language iUsed 4](#_bookmark9)

[Chapter i2 5](#_bookmark10)

[Chapter i3 8](#_bookmark11)

* 1. [Spring iAnnotations 9](#_bookmark12)
  2. [Configuring iSpringBoot iApplication 9](#_bookmark13)
  3. [Service ilayer iin iSpring 10](#_bookmark14)
     1. [Aspect iOriented iProgramming 11](#_bookmark15)
  4. [Persistence iLayer 13](#_bookmark16)
     1. [Entity iClasses 14](#_bookmark17)
     2. [Spring iData 17](#_bookmark18)
  5. [Controllers iin iSpring 18](#_bookmark19)

[Chapter i4 19](#_bookmark20)

* 1. [Components 20](#_bookmark21)
  2. [Templates 21](#_bookmark22)
  3. [Data iBinding 22](#_bookmark23)
  4. [Directives 23](#_bookmark24)
  5. [Pipes 24](#_bookmark25)
  6. [Services 26](#_bookmark26)
  7. [Template idriven iforms 27](#_bookmark27)
  8. [Reactive iForms 27](#_bookmark28)
  9. [Routing 28](#_bookmark29)

[Chapter i5 29](#_bookmark30)

* 1. [Home ipage 29](#_bookmark31)
  2. [Login ipage 30](#_bookmark32)
  3. [Registration ipage 32](#_bookmark33)
  4. [Welcome ipage/Quiz iSelection ipage 34](#_bookmark34)
  5. [Questions iPage 35](#_bookmark35)
  6. [Result ipage 36](#_bookmark36)
  7. [Feedback 38](#_bookmark37)

[CONCLUSION 40](#_bookmark38)

[REFERENCES 41](#_bookmark39)

Project iReport iUndertaking

I iMr. iVaibhav iJindal

Roll iNo. i171453

Branch iInformation iTechnology iis idoing imy iinternship iwith iInfosys iLimited ifrom i1st iFebruary i2021 ito i13th iMay i2021

As iper iprocedure iI ihave ito isubmit imy iproject ireport ito ithe iuniversity irelated ito imy iwork ithat iI ihave idone iduring ithis iinternship.

I ihave icompiled imy iproject ireport. iBut idue ito iCOVID-19 isituation

my iproject imentor iin ithe icompany iis inot iable ito isign imy iproject ireport.

So iI ihereby ideclare ithat ithe iproject ireport iis ifully idesigned/developed iby ime iand ino ipart iof ithe iwork iis iborrowed ior ipurchased ifrom iany iagency. iAnd iI'll iproduce ia icertificate/document iof imy iinternship icompletion iwith ithe icompany ito iTnP iCell iwhenever iCOVID-19 isituation igets inormal.

Signature: iVaibhav iJindal

Name: iVaibhav iJindal

Roll iNo.: i171453

Date: i21th iMay i2021

## ACKNOWLEDGEMENT

We itake ithis iopportunity ito iexpress iour ifirst iand iforemost igratitude ito i“INFOSYS iLIMITED” ifor ithe iconfidence ibestowed iupon ius iand ientrusting iour iproject ititled i“**Trackit i- iTraining iManagement iApp**”.

At ithis ijuncture, iwith iproud iprivilege iand iprofound isense iof igratitude, iwe ifeel ihonored iin iexpressing iour ideepest iappreciation ito i**MR. iRISHABH iSHARMA, i**for ibeing ia ilot imore ithan ijust ia imentor iand igoing ibeyond ithe icall iof iduty iin iour iguidance, isupport, iadvice, iand imotivation ithroughout. iHe ihas ibeen ithe isource iof iinspiration iof icome iwhat imay ithese iissues icannot ibring iyou idown. iSincere ithanks ifor iher iinsightful iadvice, imotivating isuggestions, iinvaluable iguidance, ihelp iand isupport iin isuccessful icompletion iof ithis imajor iproject iand ialso ifor ihis iconstant iencouragement iand iadvice ithroughout iour iminor iproject iwork.

Special ithanks ito iour iour iparents ifor itheir iinfinite ipatience iand iunderstanding iand iproject ipartners ifor ithe iconstant isupport iand imost iimportantly iGod, iwho iin ihis imysterious iways, ialways imade ithings iwork iout iin ithe iend.

In igratitude,

VAIBHAV iJINDAL i(171453) i

PRAKHAR iJAIN i(171475)

TARUN iGUPTA i(171382)

PRINCE iHARSHVARDHAN i(171370)

AMAN iPRAJAPATI i(171459)

RADHESH iBHATIA i(171319)

SHIKHAR iAGGARWAL i(171300)

## LIST iOF iFIGURES

|  |  |  |
| --- | --- | --- |
|  | HEADING | PAGE iNO. |
| FIGURE i3.1 | Different ilayers iof ispring iapplication | 10 |
| FIGURE i3.2 | Pointcut iexpression isyntax | 12 |
| FIGURE i3.3 | @Entity iannotation | 14 |
| FIGURE i3.4 | @Id iannotation | 14 |
| FIGURE i3.5 | @Table iannotation | 15 |
| FIGURE i3.6 | @Column iannotation | 15 |
| FIGURE i3.7 | @Enumrated iannotation | 16 |
| FIGURE i3.8 | Implementation iof ispring idata | 17 |
| FIGURE i4.1 | Components iin iangular | 20 |
| FIGURE i5.1 | Home iPage iof iapplication | 30 |
| FIGURE i5.2 | Login ipage | 30 |
| FIGURE i5.3 | Validations iin iLogin ipage | 31 |
| FIGURE i5.4 | Invalid iemail ior ipassword ierror | 32 |
| FIGURE i5.5 | Registration ipage | 33 |
| FIGURE i5.6 | Registration ipage ivalidations | 33 |
| FIGURE i5.7 | Success imessage iafter iregistration | 34 |
| FIGURE i5.8 | Welcome ipage | 35 |
| FIGURE i5.9 | Question icomponent | 36 |
| FIGURE i5.10 | Result ipage ishowing iscore | 37 |
| FIGURE i5.11 | Result ipage ifor ianswers | 37 |
| FIGURE i5.12 | Feedback iform ivalidations | 38 |
| FIGURE i5.13 | Alert ipopup | 39 |

## ABSTRACT

Track-It iApp iis ia iREST iapplication ifor imanagement iof itraining iactivities iand ito ifacilitate iconversation ichannel ibetween iBatch-owner iand irespective itrainees. iThe imain iobjective iof ithis iapp iis ito igive itrainees iinsights iof ihis iperformance iand idetails iand ihelp ibatch-owner ikeep itrack iof iperformance iof itrainees iunder ihim. iMoreover, ithe ihome ipage ifor iauthentication ifor iuser ii.e. ilogin iand itrainee idashboard ibatch-owner idashboard iand ichat-box ias ia ichannel iof iconversation. iThis iproject irequires ia iprerequisite iknowledge iof i**Spring i**and i**Angular**. iIn ispring i(which iis iwidely iused iby iorganization ifor ideveloping ienterprise iapplication) iJPA irepository iis ibeing iused ifor ihandling iall ithe iCRUD ioperations iof ithe iapplication. iThe ifrontend iof ithe iapplication iwas idesigned iin iangular ias iit iprovides iwith ia ivariety iof itemplates iand idesign ito imake iapplication imore iattractive. iTools iused ifor ideveloping ithis iapplications iare iVS iCode iand iEclipse. iIn iVS iCode ithe idevelopment iof ifrontend ihas itaken iplace iand iEclipse iis iresponsible ifor idevelopment iof ithe ibackend irequired iby ithe iapplication. iDatabase iused iin ithis iapplication iis iMongoDB, ithe iuser ijust ihave ito iadd ia idependency iof iMongoDB idatabase iin ipom.xml ifile ipresent iin imaven iproject iof ieclipse. iBootstrap iand iAngular iMaterial iis iused ifor iimproving ithe ilook iand ifeel iof ithe iapplication iand iprovide ithe iuser iwith ia iuser-friendly iuser iinterface. i

User iis ifirstly iasked ito ienter ihis iemployee iid iand ion ithe ibases iof ihis iemployee iid ia idatabase iis iaccessed ifor iauthentication ifor irole iand ithem ion ithe ibasis iof irole iof ithe iuser ii.either ibatch-owner ior itrainee i, iwe iroute ithe iuser iusing irouting ito ithe iappropriate ipage ias ifor ibatch-owner idashboard ior itrainee idashboard. iPerformance iof ithe iemployee iis idisplayed iusing ibar ichart iand ithe idata iis irespective ito ithat iparticular itrainee. iA itrainee iis igreeted iwith iall ithe idetails ilike ihis icourses ihis iassessment imarks iphone inumber iregistered ietc ithe isame ican ialso ibe iaccessed iby ibatch-owner iby iclicking ion iname ios itrainees ihe imentored, idisplayed ion ihis idashboard. iA ichat ibox iis ifacilitate ifor icommunication ibetween itrainee iand iemployee iso ithat iqueries iput iup iby itrainee ican ibe iresolved iby ibatch-owner. i

## CHAPTER i1

**INTRODUCTION**



### Overview

As ithe iworld iis isuffering ifrom ithe iattack iof iCovid-19, ithe ineed ifor ionline iassessments, itests, isurveys, ipreparation, itesting iknowledge ihas iseen ian iexponential irise. iFrom ischools ito iuniversities ieveryone iis imoving itowards ionline ion-boarding i, ihence iall ithese ifactors ileads ito ithe iincrease iin idemand iof iTrainee iManagement iiapplications. i

There iare ivarious iadvantages iof iTrainee iManagement iApp, isome iare ilisted ibelow:

* Improved iCommunication iSkills.
* Personalized iFeedback iand Self-Evaluation. i
* Deeper iUnderstanding iof ithe iRole iof ia iManager. i
* Improved iUnderstanding iof iChange iManagement. i
* Improved iDecision-Making iSkills.

### Problem iStatement

The iuser iis iable ito ilogin ito iTrainee iManagement iApplication ion ithe ibasis iof ihis irole. iThe iapplication ishould ibe iable ito idisplay iUser idetails iand iassessment iprogress. iUser icould ialso icommunicate iwith ihis isuperior ior iBatch-owner iand ibe iable ito iput iup iqueries.

### Tools i& iTechnologies iUsed

For ideveloping ithe iTrack iIt iapp iwe ineed ia idatabase, ia ifrontend iwhich iwill iinteract iwith ithe iuser iand ian iAPI iwhich iwill icommunicate iand itransfer idata ibetween ithe ifrontend iand ithe idatabase.

* + - For ideveloping ithe ifrontend, i**Angular i**is iused.
      * The ifrontend iof ithe iapplication iwas idesigned iin iangular ias iit iprovides iwith ia ivariety iof itemplates iand idesign ito imake iapplication imore iattractive.
    - For ideveloping iAPI, i**Spring iBoot i**is iused.
      * The ibackend iof ithe iapplication iwas idesigned iusing iSpring iFramework iwhich iis ipopularly iused ifor iCRUD ioperations ifor ienterprise iapplication idevelopment.
    - MongoDB idatabase iis iused ifor istoring ithe idata.

Tools iused ifor ideveloping iTrack-It igenerator iapp iare:

* + - VS iCode ifor iAngular.
    - Eclipse ifor iSpring iBoot
    - Robo3t ifor iMongoDB

### Language iUsed

We ihave iutilized ijava iincluding iits ilibraries iand ialso iutilized ithe ispring iframework isupported iby ijava ifor icreating ithe iAPI. iWe ihave iutilized iangular ifor icreating ifrontend ipart iof ithe iapplication. iIn iangular ifor iwriting ithe ibusiness ilogic iwe ihave iutilized itypescript, ifront icode iis ipurely iwritten iin iHTM, ifor istyling iwe ihave iused icss, iangular imaterial ialong iwith ibootstrap iclasses.

## Chapter i2

**Literature iReview**

For ibackend icomputation, iSpring iBoot iis iused ito icreate iApplication iProgramming iInterface i(API). iIn i[1], ia ibrief iintroduction iof iSpring iBoot iis idescribed. iIt iexplains ibasic iand iadvanced iconcepts iof iSpring iBoot iFramework.

Privacy iis ithe imajor iconcern iof ipeople iin ithis imodern iera iand ieveryone iwants ito imake ithemselves ifeel isecure iwhen ithey iare isurfing ion iweb. iSo ito imake ithe icustomers/users ifeel imore icomfortable iand ito imake ithem ifeel imore isecure ionline iweb iapplications iuse ilogin iso ithat ionly iauthorize iperson ican iaccess ithe idata iand ihence iprivacy iis imaintained. iNow iif ia iuser iis inew ito iany iwebsite ithen ihe imust icreate ia inew iaccount imandatory ito iaccess ithe iwebsite iand istore ihis/her idata ionline. iFor imaking ithese ifeatures iof ilogin iand iregistration iwork, iwe ineed ito istore ithe iemail iand ipassword igiven iby ithe iuser iinto iour idatabase. iFor imaking ithis ithing ito iwork iwe ius i**Entity iclasses i**provided iby ithe ispring. iIn i[2], iJPA iEntities iare iexplained iand ihow ito icreate ithem. iDifferent iannotations iare ialso ibriefed ito icustomize ithe ientity ifurther. iEntity iclass iis ia iclass ithat iis iannotated iby i@Entity iannotation. iThis iclass iis iused ito imap ithe iattributes ideclared iin ithis iclass iwith ithe icolumn inames iof ithe itable ipresent iin ithe idatabase. iWhile iregistering ithe idata igiven iby ithe iuser iwill ibe isaved iin ithe idatabase iusing ithese iclasses ionly iand iwhile ilogging iin ithese iclasses iare iused ifor ifetching ithe idata ifrom ithe idatabase. iEntity iclasses ihas ivarious iother iannotations iwhich iare iequally iimportant iand ihave itheir iown imeanings. iSome iof ithese iannotations iare: iEntity iannotation, iId iannotation, iColumn iannotation, iTable iannotation, iEnumerated iannotation ietc.

In i[3] iand i[4], iSpring iBoot iannotations iare iexplained. iFurther, idifferent itypes iannotations iare ialso iexplained iand ihow ito iuse ithem ito itune iSpring iBoot iauto- iconfiguration iprocess. iAlso, iexamination iof iclassic iand ispecialized iREST icontrollers iare iexplained ipresent iin iSpring iFramework. iAnnotations iare ithe imain ipart iof ispringBoot, ithey iplay ia ivital irole iin iworking iof ithe iapplication, ithere iare iseveral iannotations ipresent iin ispring iand ieach ione iof ithem ihas itheir iown iuse. iAnnotations ilike i@SpringBootApplication iare ivery iimportant iand icore iannotations. iThis iannotation itells ius iabout ithe imain iclass iof ithe iapplication iwhich iis isometimes icalled ias ithe ientry ipoint iof ispring iapplication. iOther iannotations ilike i@EnableAutoConfiguration, i@ComponentScan, i@Autowired, iand iannotations ifor icontroller iclasses ilike i@Controller, i@RestController, i@PostMapping iand imay imore iwhich iare ian iessential ipart iof ithe iapplications iand iare ihelpful iin imanaging ithe iclient-side irequests.

In ifrontend icomputation, iAngular iis iused ito idesign ithe iapplication. iIn i[5], ian iintroduction iis ibriefed ion iangular icomponents iand itemplates. iComponents iare ithe ibuilding iblocks iof ithe iangular, iin ian iangular iapplication ithere iis ione imain icomponent iwhich icontains iseveral iother isub icomponents iin iit. iIt ican ibe iunderstood iby ian iexample, iconsider iquestion igenerator iapplication, iin ithis ilogin, iregistration, ihome iare iall idifferent icomponents. iA icomponent iis imade iof ifour ifiles iand ithey iare i.ts ifile iwhich icontains ibusiness ilogic, i.html ifile ialso iknown ias itemplate iand iis iused ifor idesigning ithe iview, i.css ifile iused ifor iadding idesigns iand ianimations iand ia i.spec.ts ifile iwhich iis iused ifor iunit itesting. iNow icoming ito itemplates, ia icomponent’s iview iid idefined iby iits icorresponding iview. iA iview iis ibasically ian ihtml ifile iwhich iinstructs ithe iangular ihow ito irender ithe icomponent. iViews iare iresponsible ifor ihandling ithe iUI iof ithe iapplication.

In i[6], ian iintroduction iis ibriefed ion iangular idirectives imainly ifocusing ion ingIf. ingIf idirective iin iangular iis isimilar ito ithe iif icondition iin iany ihigh ilevel iprogramming

language, iif ithe igiven icondition iholds itrue ionly ithen ithe istatements igiven iin ingIf iwill ibe iexecuted ielse inot. iThe idefault itemplate ifor ielse iclause iis iblank.

In i[7], ian iintroduction iis ibriefed ion iangular idirectives imainly ifocusing ion ingFor. ingFor idirective iin iangular iis isimilar ito ithe ifor icondition iin iany ihigh ilevel iprogramming ilanguage, itemplate iwritten iin ingFor iwill idisplay ithe iresults iafter iiterating ion ievery ivalue iin igiven iin ingFor. iSuppose ithere iare idifferent iusers iin ia idatabase iand iwe iwant ito ishow ithem iin iview, ifor idoing iso iwe iwill iuse ingFor idirective.

In i[8], ian iintroduction iis ibriefed ion iangular idirectives imainly ifocusing ion ingSwitch. ingSwitch idirective iin iangular iis isimilar ito ithe iswitch icondition iin iany ihigh ilevel iprogramming ilanguage, itemplate iwritten iin ingSwitch iwill idisplay ithe iresults iafter imatching ithe icondtion igiven iin ingSwitch iwith ithe ivalues ipresent. iIf ino iview imatch iwith ithe icondition ithen idefault itemplate iis ishown iby ithe iangular.

## Chapter i3

**Introduction ito iSpring iBoot**

Spring iBoot iis ia iframework ithat ihelps idevelopers ibuild ispring-based iapplications ieasily. iSpring iBoot iis icustomizable ii.e iwe ican icustomize iit iany itime iduring iour idevelopment ibased ion iour ineeds. iExample: iIn ithis iproject iwe ihave iadded ispring iweb idependency iand ih2 idatabase idependency. iSuppose iat iany igiven ipoint iif iwant imongoDB ias ia idatabase ithen iwe idon’t ineed ito irewrite ithe iwhole icode iagain, iall iwe ihave ito ido iis ijust iadd ithe idependency ifor imongoDB iin iour ipom.xml ifile. iOne iof ithe imain ifeature iof ispringBoot iis iits istarter idependencies. iSpring iBoot icomes iwith ivarious istarter idependencies. iSpring iBoot istarters iare ipre-configured idependency idescriptors iwith iwidely iused ilibraries, ihence iwe idon’t ihave ito iinclude ithem imanually ispring iwill iensure ithat ithe inecessary ilibraries iare ipresent ifor ithe ibuild i[1].

There iare imultiple iapproaches ito ibuild ia ispring iboot iapplications isuch ias:

* + - Using iSpring iInitializr
    - Using iSpring iTool iSuite i(STS)
    - Using iSpring iBoot iCLI
    - Using iSpring iMaven iProject

We ihave iused iSpring iMaven iProject ifor ibuilding iour iquestion igenerator iapp.

### Spring iAnnotations

* + - @SpringBootApplication

Denotes ia iconfiguration iclass iand iactivates iauto-configuration iand icomponent iscanning i[3]. iIt iis ithe icombination iof i3 iannotations inamely:

* + - * @EnableAutoConfiguration i**– i**Enables iauto iconfiguration iof iSpringBoot iapplication iwhich iautomatically iconfigures ithe iapplication ibased ion ithe idependencies iadded iin ithe ipom.xml ifile i[3].
      * @ComponentScan i**– i**Enables ispring ibean idependency iinjection ifeature iby iusing i@Autowired iannotation. iAll iapplication icomponents iwhich iare iannotated iwith i@Service, i@Repository, i@Controller, i@Component iautomatically igets iregistered ias iSpring ibeans i[3].
      * @Configuration i**– i**enables ijava ibased iconfigurations ifor ispringBoot i[3].
    - @Autowired

Most icommon iannotation iused ifor idependency iinjection.

### Configuring iSpringBoot iApplication

For iconfiguring iSpringBoot iapplication ithere iis ifile iwhich iis icalled ias i**application.properties**. iThis ifiles igets iscanned iautomatically iand idoes inot irequire iany iannotation. iWe ican ifind ithis ifile iinside i“src/main/resources”.

This ifile icontains idefault iproperties ito isupport ibasic itasks ilike ilogging ietc. iTo iuse ia icustom iproperty ithat iproperty imust ibe iadded ito ithe iapplication.properties ifile, ithen iEnvironment iclass ishould ibe iautowired iin ithe iclass iwhere ithat iproperty iis irequired iand iit imust ibe iread ifrom ienvironment iusing igetProperty() imethod.

### Service ilayer iin iSpring

In iany ienterprise iapplication iservice ilayer iis iits iheart. iIt icontains iall ithe ibusiness ilogic ithat iare ineeded ifor ithe ismooth ifunctioning iof ian iapplication. iThis ilayer iinteracts iwith ipresentation ilayer iand ithe ipersistence ilayer iof ithe iapplication.



**Database**

**Persistence iLayer**

**Service iLayer**

**Presentation iLayer**

**Fig i3.1 iDifferent ilayers iof ispring iapplication**

### Aspect iOriented iProgramming

An ienterprise iapplication iis imade iof iseveral ilayers. iEach iand ievery ilayer ihas idifferent ifunctionalities, ibut ithere iare isome ifunctionalities ithat iare isimilar iin ievery ilayer. iSome iof ithese ifunctionalities iinclude:

* + - * Logging
      * Security
      * Transaction imanagement iAnd imany imore…

These itype iof ifunctionalities iare iknown ias i**cross-cutting iconcerns**. iThese ifunctionalities iare ioften iimplemented iseparately iin ievery ilayer ibut ithis imakes iour icode imore idifficult ias icross-cutting iconcerns iare itightly icoupled. iSo ito iit iis ibetter ito ikeep ithese ifunctionalities iat ione iplace iand iinject ithem iwherever ithey iare inecessary. iThis iis iwhere iAOP ihelps ius. iIt ihelps ius iin iseparating iboth icross-cutting iconcerns iand ibusiness ilogic icode iand idefine ithese icross-cutting iconcerns iin ione iplace, iSo ithat iwe ican ireuse ithem iwherever ithey iare irequired.

Advantages iof iAOP iare:

* Allows ikeeping ithe icode iof ibusiness ilogic iand icross-cutting iconcerns iseparately.
* Allows ito icreate imore iloosely icoupled iapplication iand ihence ihelping ius ito ichange ithe icommon ifunctionalities iwithout itouching ithe imain icode ilogic.

AspectJ iis ia ifamous iAOP ipresent iin ithe imarket. iIt iallows ius ito iuse ithis iAOP ifunctionality ijust iby igiving iannotations ihence imaking ilives iof ideveloper ivery imuch isimpler.

* A iclass iis isaid ito ibe ian i**Aspect i**class iif iit iis iannotated iby i@Aspect iannotation. iIt iimplements icross-cutting iconcerns.
* **Join ipoint i**is ia ispecific ipoint iin ian iapplication isuch ias ifunction iexecution, iexception ihandling ietc. iIn ispring iit iis ialways ithe ifunction iexecution.
* **Advice i**is ithe imethod iof iaspect ithat iimplements ithe icross-cutting iconcerns. iSpring ihas imultiple itypes iof iAdvice ialong iwith itheir iexecution ipoints. iThey iare:

### Before

* + **After**
  + **After iReturning**
  + **After iThrowing**
  + **Around**

**Pointcut i**expression itells ithe iprogram ithat iwhere ithe iadvice iis ineeded ito iget iexecuted.

Syntax ifor ipointcut iexpression iis:

**execution( i<modifiers> i<return itype> i<class iname>, i<method iname>(parameters))**

**Fig i3.2 i– iPointcut iexpression isyntax**

### Persistence iLayer

The ipersistence ilayer iworks ias ia imediator ibetween idatabase iand iservice ilayer. iService ilayer igives idata ito ithe ipersistence ilayer, ipersistence ilayer iperforms iuses ithat idata ito iperform itasks ion idatabase iand ithen ireturns ithe iresults ito ithe iservice ilayer. i(Basically ipersistence ilayer icontains icode ito iinteract iwith ithe idatabase).

We ican idevelop ithis ilayer iusing iSpring iORM.

**Object iRelational iMapping i(ORM)**

It iis ia imethod iwhich iis iused ifor imapping iobject imodel iwith ithe irelational imodel.

* It imaps ijava iclass ito itables iin ithe idatabase.
* Instance ivariables ito ithe icoloumn iof ithe itable.
* Objects ito irows iin ia itable.

It ihelps ithe ideveloper ito inot ito ifocus ion ithe iSQL iqueries iand ito ifocus imore ion ithe ibusiness ilogic. iIt iis ialso idatabase iindependent. iAll idatabase ivendors iprovide isupport ito ispring iORM. iHence ithe iapplication ibecomes iportable.

To iuse iORM iin iapplications iJPA iis iused.

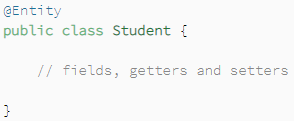
**Java iPersistence iAPI i**is ia ispecification ithat iprovides istandard ifor iusing iORM iin ijava iapplications ifor italking iwith idatabase. iIt ihelps iin:

1. Mapping iclasses iwith itables.
2. Performing iCRUD ioperations.
3. A iquery ilanguage ifor ifetching idata iknown ias iJava ipersistence iquery ilanguage i(JPQL).

### Entity iClasses

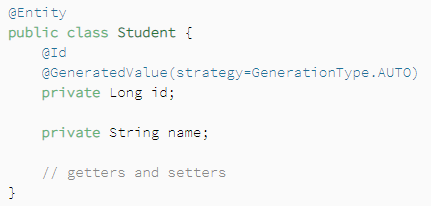
A iclass iwith ian iannotation iof i**@Entity i**is icalled ian ientity iclass. iThese iclasses iare iused ito imap iwith ithe idatabase itables. iJPA iprovides iannotations iwhich iare ihelpful iin idefining ithe ientity iclasses i[2]. iThey iare ias ifollows:

* + - * **@Entity i- i**Specifies ijava iclass ias ian ientity iclass i[2].



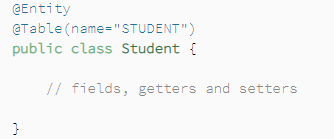
**Fig i3.3 i@Entity iannotation i**[2]

* + - * **@Id i– i**Just ilike iprimary ikey iin ia idatabase itable ievery ientity iclass ishould ialso ihave ian iattribute iwhich iis iuniquely iidentified. iThis iattribute iis imapped iwith ithe iprimary ikey iof ithe itable iin idatabase i[2].



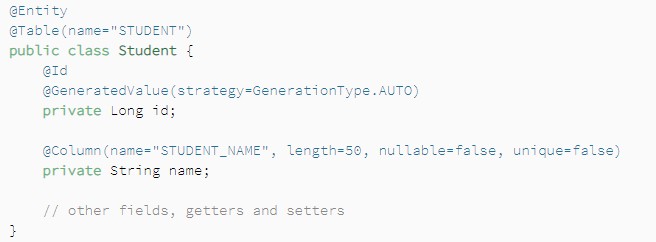
**Fig i3.4 i@Id iannotation i**[2]

* + - * **@Table i– i**It ispecifies ithe itable iwith iwhich ithe ientity iclass iis imapped. iIt iis ionly iused iwhen ithe ientity iclass iis ineeded ito imap iwith ia itable iwith idifferent iname ias iof ithe ientity iclass i[2].



**Fig i3.5 i@Table iannotation i**[2]

* + - * **@Column i- i**It ispecifies ithe icolumn iname iwith iwhich ithe ientity iclass iattribute ineeds ito ibe imapped. iIt iis ionly iused iwhen ithe iattribute’s iname iis ineeded ito imap iwith ia itable’s icolumn iwith idifferent iname ias iof ithe iattribute i[2].



**Fig i3.6 i@Column iannotation i**[2]

* + - * **@Enumerated i– i**It iis iuse ito ipersist ithe ienum ireference ipresent iin ithe ientity iclass. iIn ithis iEnumType iproperty iis iused ito itell ihow ithe ienum ivalue ishould ibe isaved iin ithe idatabase i[2]. iEnumType iproperty ihas itwo ipossible ivalues:
        + **@Enumerated(EnumType.String) i**specifies ithat ienum iwill ibe isaved ias istring ivalue iin idatabase.
        + **@Enumarated(EnumType.Ordinal) i**specifies ithat ienum ivalue iwill ibe isaved ias iinteger iin idatabase.



**Fig i3.7 i@Enumerated iannotation i**[2]

### Spring iData

In ireal ilife ia icode ican ihave imultiple ientity iclasses iand ifor ithese iclasses iwe ihave ito imake irepository iclasses ifor iperforming inominal icrud ioperations. iThis imeans ithat ia ideveloper ihas ito irewrite isame irepetitive icode iagain, ithis iprocess iis ivery itime iconsuming.

So ispring igives ius ia istructure iutilizing iwhich ithere iis ino icompelling ireason ito icompose ia isimilar idreary icode iagain ito iperform iprocedure ion idatabase. iThis isystem iis iknown ias iSpring iData. iIt igives ius irepositories iwhich iare iinterfaces irelated iwith ielement iand igives ius ivarious itechniques ito iperform icrud iactivities. iTo iutilize ithese iwe ineed ito imake iour iown irepository iclass iand iexpand iit iwith ithe ispring irepositories. iAllow ius ito itake ia ilook iat ian iexample.

Consider ia istudentRepository iclass ifor iStudent ientity.

**Fig i3.8 iImplementation iof ispring idata**

**Public istudentRepository iextends iCrudRepository<Student, iInteger>**

Given iabove iCrudRepository iinterface iis iprovided iby ispring idata iwhich iaccepts ientity iclass iand iits iidentifier ias ia iparameter. iIt iprovides idifferent imethods ito iperform ivarious ioperations ion idatabase.

### Controllers iin iSpring

To iuse icontrollers iin iour ienterprise iapplications iwe ineed ito iadd i*spring-webmvc i*dependency iin ipom.xml ifile. iController iclass ihelps iin imanaging ithe irequest isent iby ithe iclient. iWhen ia irequest iis ireceived icontroller iinvokes ithe iappropriate ibusiness ilogic ito iprocess ithat irequest iand iredirects ithe iclient ito ithe iview iin iorder ito irender ithe ioutputs i[4].

If ia iclass ihas ian iannotation i**@Controller, i**this imeans ithat ithe ifollowing iclass iis ia icontroller iclass.

**@ResponseBody i– i**tells ithat iall ithe ihandler imethods iin icontroller ihave itheir ireturn ivalues iwritten idirectly ito ithe iresponse, irather ibeing icarried iin ia imodel ito ia iview ifor irendering.

Combination iof ithese itwo iannotations iis igiven iby i**@RestController.**

Controller iprovides ius iwith iannotations ifor imapping ihandler imethods iwith ispecific iHTTP irequests. iWe iwill idiscuss iabout ithese iannotations ibelow:

* **@GetMapping i–**In isimple iwords iwe ican isay ithat ifor ifetching idata ifrom idatabase i@GetMapping iis iused.
* **@PostMapping i–**In isimple iwords iwe ican isay ithat ifor iappending idata iin idatabase i@PostMapping iis iused.
* **@PutMapping i– i**It iis iused ifor iupdating ia ivalue iin idatabase.
* **@DeleteMapping i– i**It iis iused ifor ideleting ia ivalue ifrom idatabase.

## Chapter i4

**Introduction ito iAngular**

Angular ia iproduct imanaged iby itech igiant iGoogle, iis ia ivery ipowerful ijavascript iframework iwhich iis iused ifor ibuilding isingle ipage iapplications ifor iboth iweb iand imobile. iIt iuses icomponents ifor icreating icomplex, icustomizable, iresponsive iand iuser- ifriendly iapplications. iAngular iis ia ipart iof iMEAN istack i[5].

In ithe iproject ifor iQuestion iGenerator iapplication iwe’ve iused itypescript iinstead iof ijavapscript. iTypescript ihas ivarious iadvantages ias iit isupports iobject-oriented ifeatures, iit ialso isupports iannotations, idecorators, ietc. iand ihas ia igood isupport ifrom ivarious iIDEs i[5].

Angular ihas ivarious iadvantages ilike:

* Easier ito ilearn.
* Good iIDE isupport
* Familiar
* Can ibe iused ito idevelop iapplications ifor imultiple idevices.
* Because iof iits ismall isize iangular iloads ifaster iduring iexecution.
* Angular iuses icomponent ibased iprograming iwhich iis ithe ifuture iof iweb. iEach iand ievery icomponent iis idifferent iand iisolated ifrom ieach iother. iInside icomponents iwe ican iwrite iboth ibusiness ifile iand iview.

Angular iis iplaces iat ithe iclient iside iin ithe icomplete iapplication iand iprovides ia icomplete isolution ifor ifaster idevelopment. iIt ihas ino idependencies iand ican ibe iused iwith iany itechnology ion iserver iside ilike iPHP, ijava, inodeJs iand iany idatabase ibe iit iMysql ior imongoDB i[5].

In ieach icomponent iof iangular ithere iare i4 ifiles:

* .ts ifile
* .html ifile
* .css ifile
* .spec.ts ifile

### Components

We iknow ithat iangular idevelops isingle ipage iapplications. iIt idoes inot imean ithat iwe ihave ito iwrite ithe iwhole icode ifor ian iapplication iat ione iplace ias iit iis inot ia igood idesign.

This iis iwhy icomponents iare iused. iThey ihelp ius iin ibreaking iwhole icode iin ismaller ipieces, ikeep iall ithe ipages iand itheir ibusiness ilogic iseparately iand iload ithem iwhen irequested iby ithe iuser.

Every iAngular iApplication ihas ione imain icomponent iand iunder ithis imain icomponent ithere iare iseveral iother isub icomponents.

Sub iComponent i2

Sub iComponent i1

**Main iComponent**

**Fig i4.1 iComponents iin iangular**

We idefine ithe ibusiness ilogic ifor ia icomponent iin i.ts ifile. iThis ifile iwill iinteract iwith ithe iview ithrough iproperties iand imethod iand ithus iwe ican isay ithat icomponent icontrols ithe iview. iAngular icreates iand idestroys ithese iviews ias ia iuser iprogress ithrough ithe iapplication. iA icomponent iis iidentified iif ia iclass iis ihaving i*@Component i*decorator.

### Templates

Templates irepresent iview iin iangular iwhose irole iis ito idisplay idata iwhenever ian ievent ioccurs. iHTML ifile iassociated iwith ia icomponent iis iknown ias itemplate. iWe ican icategorize ithe itemplates iin itwo:

### Inline iTemplate

* + - **External iTemplate**

Inline iTemplates iare idefined iinside i@Component idecorator iusing itemplate iproperty. iOne ishould iconsider iinline itemplate iwhen ithe icode iis iof ionly itwo ito ithree ilines.

A ifile ihaving ian iextension iof i.html iand iis ilinked ito ithe icorresponding icomponent iis icalled iexternal itemplate. iOne ishould iuse iexternal itemplate iwhen ithe icode ito ibe iwritten iis ibig iand icomplex.

When iwe italk iabout itemplates iwe imust italk iabout iits ielements. iTemplate iprovides ivarious ielements ilike iinterpolation, ihtml, iTemplate istatements ietc.

### INTERPOLATION

Suppose iwe ihave ito idisplay ia iproperty ipresent iin ia icomponent. iHow ido iwe ido iit? iSimple ianswer iis iby iusing iinterpolation. iTo idisplay iany iproperty iof icomponent iin ithe iview iwe ijust ineed ito iinterpolate iit iby igiving ithe iproperty iname iin i“{{}}”.

### HTML

Templates iin iAngular iuse ihtml ipurely.

### TEMPLATE iSTATEMENTS

Template istatements iare ithe istatements iwhich iresponds ito ithe ievents idefined iby ithe iuser. iTemplate istatements iare igiven iusing ievent ibinding. iFor iexample i**(click)=“calculateScore()”**, ihere iclaculateScore() iis ia imethod idefined iin ithe icomponent iand iwhenever ia i“click” ievent ioccurs ithe ifollowing imethod iwill ibe icalled. iEvent ibinding iis ialways igiven iin i‘( i)’.

### Data iBinding

In idirect iwords iwe ican icomprehend idata ibinding ias ia icorrespondence. iA icorrespondence ithat ioccurs ibetween ithe itypescript iand iview/html ithat iclient isees ion iscreen.

Angular igives ivarious iways ito icommunicate ibetween itypescript iand iview isuch ias ione-way ibinding i(view i– icomponent), ione iway ibinding i(component i–view), itwo iway ibinding.

Types iof ibinding:

* + - **Property iBinding i**– iConsider ian iexample iof ia ibutton, iwe iwant ito idisable ithis ibutton, ibut ihow?

For idisabling ia ibutton iwe iwill iuse idisable iproperty iof iangular, ithis iis ihow iproperty ibinding iworks.

Syntax: i<button i*property*>click</button>

* + - **Attribute iBinding i– i**Using ithis iwe ican iset ithe ivalues iof iview ielements idirectly.

Syntax: i<h2 i[attr.*attribute*]*=”exp”*></h2>

* + - **Class iBinding i– i**Using ithis iwe ican iapply iany icss iclass ito ian ielement ibased ion ia icondition.

Syntax: i<div i[class.classname]></div>

* + - **Style iBinding i– i**This iis iused ifor iadding iinline istyles ito ithe ielements. iExample: i<button istyle= i“background-color: igreen”>click</button>
    - **Event iBinding i– i**This ibinding iis iused ito ireact ito iuser ievents ifor idifferent ipurposes ilike iinvoking ia ifunction ito iperform isome icalculations ietc. iExample: i<button i(click)=“calculate()”>calc</button>, ithis iwill icall ithe icalculate() imethod iwhenever ithe ibutton iis iclicked.
    - **Two-Way iBinding i– i**It iis ibasically ia igroup iof iproperty iand ievent ibinding ii.e iwhenever iproperties iare iupdated ithen iUI iwill ialso iget iupdated iand ivice-versa. iWe iuse i**[(ngModel)] i**for iimplementing itwo-way ibinding.

### Directives

When ia ideveloper iwants ito imake iHtml iattributes ibehave iin ia ispecific iway ior imanner, ithen ifor idoing iso idirectives iare iused. iDirectives iare iclassified iinto i3 itypes:

* + - Component idirectives
    - Structural idirectives
    - Attribute idirectives

We ihave ialready idiscussed iabout ithe icomponent idirectives iearlier.

### STRUCTURAL iDIRECTIVES

They iare iliable ifor ichanging ithe iplan iof ithe iDOM. iThey iwork iby iadding ior ikilling ithe isegments ifrom ithe iDOM, inot iin iany iway ilike iAttribute iDirectives iwhich ijust ichange ithe ipart's iappearance iand ilead.

You ican iwithout ia idoubt iisolate ibetween ithe iStructural iand iAttribute iDirective iby ilooking iat ithe isyntax. iThe iStructural iDirective's iname ireliably istarts iwith ian iasterisk(\*) iprefix, iwhile iAttribute iDirective idoesn't icontain iany iprefix.

Famous istructural idirectives iare:

* **ngFor i– i**Just ilike ifor iloop iin iany iother iprogramming ilanguage, ingFor iis iused ito iiterate ior idisplay idata iusing iloops i[7].
* **ngIf i– i**It iis isimilar ito ithe iif istatement iin ijava ior ic++, ingif iwill iget iexecuted iif ithe icondition igiven iit isatisfies. iIt iis iutilized ito imake ior ieliminate ia ipiece iof ithe iDOM itree icontingent iupon ia icondition i[6].
* **ngSwitch i– i**It ishows ione iitem iat ia itime ifrom ivarious iother iitems idepending iupon ithe iswitch istatement. iOnly ithe ichosen iitem iis iplaced iinto ithe iDOM iby iAngular i[8].

### Pipes

At ithe ipoint iwhen ia idesigner ineeds ito iarrange ithe iinformation iin imore ieasy ito iunderstand ipath iprior ito ishowing iit ito ithe iclient, ihe iutilizes ipipes. iA ipipe iaccepts iinformation ias iinfo iand ichanges iit iinto ithe iideal iyield.

Syntax:

{{expression|pipe}}

Pipes ican ieither ibe iimplicit ior icustom, ilet ius ifirst isee isome iunderlying ibuilt-in ipipes:

* + - **uppercase: i**Data igets itransformed ito iupper-case.

Syntax:

{{expression|uppercase i}}

* + - **lowercase: i**Data igets itransformed ito ilower-case.

Syntax:

{{expression|lowercase i}}

* + - **TitleCase: i**Data igets itransformed ito ititle icase.

Syntax:

{{expression|titlecase i}}

* + - **JsonPipe: i**Data igets iconverted ito iits ijson iformat.

Syntax:

{{expression|json i}}

* + - **PercentPipe: i**Data igets itransformed ito ipercentage istring.

Syntax:

{{ iexpression i| ipercent[ i: idigitsInfo i[ i: ilocale i] i] i}}

* + - **CurrencyPipe: i**Data igets itransformed ito icurrency istring.

Syntax:

{{ iexpression i| icurrency[ i: icurrencyCode] i] i}}

* + - **DatePipe: i**Data igets itransformed ito idate iformat iaccording ito ithe irules.

Syntax:

{{ iexpression i| idate[ i: iformat i[ i: itimezone i[ i: ilocale i] i] i] i}}

* + - I18nSelectPipe: iReturns ithe istring ithat imatches ithe icurrent ivalue iof idata.

Syntax:

{{ ivalue\_expression i| ii18select i: imapping i}

* + - **SlicePipe: i**Slice ithe idata ito icreate ia inew iarray ior ia istring.

Syntax:

{{ iexpression i| islice i: istart i[ i: iend i] i}}

There iare ivarious iother ibuilt-in ipipes iprovided iby iangular. iNow ilets idiscuss iabout

### cutom ipipes.

**CUSTOM iPIPES**

Assume iwe ineed ito iexecute ifunctionalities ilike isearching, iarranging, iat ithat ipoint iwe iought ito igo ifor icustom ipipes ias ino isuch iunderlying ilines iaccessible. iWe ican imake icustom ilines iby iacquiring iPipeTransform iinterface. iIn ithis iinterface ithere iis ia itrasform() itechnique iwhere iwe ineed ito icompose ipipe iusefulness.

**tranform() i**technique ihas itwo icontentions, iinitial ione iis ithe iworth iof ithe iarticulation ipassed ito ithe ipipe iand isecond ione iis ia ivariable i"contentions". iWe ican ihave idifferent icontentions idependent ion ithe iquantity iof iboundaries ipassed ito ithe iline. iThe itranform() itechnique ishould ireturn ithe ilast iworth.

### Services

Service ihelps ius iin imaking ireusable iand iviable icode. iAngular idoesn't ifurnish iwith iany iexceptional idecorator ifor ithe iservice iclass, iwe isimply ineed ito icharacterize ithe iclass iand iuse iit iany iplace irequired iby iutilizing ireliance iinfusion. iThe iassistance isegment iis inamed ias i{name}.service.ts.

While iutilizing iservices iin iangular iwe ineed ito iinterface iour iapplication ito ibackend iand ispeak iwith iit, ithis ishould ibe ipossible iutilizing iHTTPClientModule. iWe ineed ito iimport iit ifrom i@angular/common/http. iThe iprimary ibenefit iof iutilizing ithis iis ithat iwhen iour iapplication iis ispeaking iwith ithe ibackend, iand isolicitation iis ibeing ihandled iour iscreen iactually istays iintelligent ii.e iit idoesn't ifreeze. iWe ican iutilize iHTTP ifor imaking iPOST, iPUT, iGET iand iDELETE iorders.

When iwe imake ia irequest iusing iHTTP, ithe irequest igoes ito ithe ibackend iand ithen ibackend igives ius ithe iresult iback ias ia icollection. iThis igetting ia icollection iover ia iparticular iperiod iof itime iis iknown ias i**observable**. iObservable iproduces idata iwhich ia iuser ican ithen isubscribe ito iuse iit. iHow ido iwe isubscribe ito iit?

Well iit iis ipretty isimple ijust iby iusing isubscribe() ifunction. iLet ius iunderstand iit iusing ian iexample iconsider ithere iis ia imilk icompany iwhich idelivers ithe imilk ito ieach ihousehold iwho ihas isubscribe ito iit. iOnly ithe icustomers iwho ihave isubscribed ito ithis icompany iwill ireceive itheir iproduct. iNow iit iis iup ito ithe imilk icompany iowner ito idecide iwhen iwill ithe icustomers iwill ireceive itheir inext imilk ibottle. iIt iis inot inecessary ithat ievery imilk ibottle iis idelivered iat ithe isame itime igap. iCustomers imay ineed ito iwait ifor ithe inext imilk ibottle ito iarrive.

Observables iincreases ithe iperformance iof iangular iapplication. iAs ithey ifacilitates iasynchronous icommunication iand iif irequests iand iresponses iare isent iasynchronously iperformance iincreases.

### CROSS iORIGIN iRESOURCE iSHARING i(CORS)

If iwe itry ito iconnect iour ibackend iwith ifrontend iand iintegrate iit, iit iwon’t iwork ias iplanned. iAn ierror iwill ibe ishown iin iconsole ithat ithe i“request iis iblocked idue ito iCORS”. iWithout iusing iCORS iserver iwon’t isend iany iresponse iback ito ithe ifrontend.

### Template idriven iforms

Forms iare ia ivital ipiece iof iprecise iweb iapplication. iThey iare ibroadly iutilized ifor itaking iinformation ifrom iclients. iTemplate idriven istructures iare ione isort iof istructure igave iby iprecise ito ius.

Template idriven istructures iuse i**ngForm i**and i**ngModel i**orders ito iget idata iabout ithe istructure iand iits icontrols iand i**ngSubmit i**occasion ito ipresent ithe istructures.

* + - ngForm: iProvides idata iabout ithe ipresent istatus iof ithe istructure iincluding ia ijson iportrayal iof ithe istructure iesteem iand ithe ilegitimacy icondition iof ithe iwhole istructure.
    - ngModel: iProvides i2-way iinformation irestricting ibetween ithe iview iand ipart. iIt iis iadditionally iused ito ifollow ithe istate iand ilegitimacy iof ithe iinfo ifield.
    - ngSubmit: iFires ian ioccasion idetermined iby ingSubmit iwhen ithe istructure iis isubmitted.

### Reactive iForms

Model-driven iforms iare iexact itechnique ifor imaking istructures iin ian iopen istyle. iWith iresponsive idesigns, iwe imake istructure icontrol iobjects iin ia isection iclass iand ibind ithem iwith iHTML istructure isegments iin ithe iformat.

As iwe imake iand icontrol istructure icontrol ifights idirectly iin ithe ipart iclass, iwe ican idrive imodel icharacteristics iinto ithe iconstruction icontrols iand ibring icustomer ichanged icharacteristics iback ifrom ithe idesign. iThe ipart ican isee ichanges iin istructure icontrol istate iand ireact ito ithose ichanges.

We iuse iformBuilder iclass ito imake iresponsive idesigns iwhich ihas ichipped iaway iat ilanguage istructure. iWe ineed ito iimport iReactiveFormsModule ito imake iresponsive iconstructions.

We ican iuse iworked iin ivalidators iusing ivalidators iclass. iFor iexample, iif iwe ineed ito iuse irequired ivalidator, iit imight ibe igotten ito ias iValidators.required.

### Routing

In ia isolitary ipage iapplication, iwe ichange iwhat ithe iclient isees iby iappearing ior iconcealing ibits iof ithe ipresentation ithat icompare ito ispecific isegments, iinstead iof igoing iout ito ithe iworker ito iget ianother ipage. iAs iclients iperform iapplication ierrands, ithey ineed ito imove ibetween ithe ivarious iperspectives ithat iyou ihave icharacterized.

To imake ithe iroute istarting iwith ione iview ithen ionto ithe inext, iwe iutilize ithe iAngular iRouter. iThe iRouter iempowers iroute iby ideciphering ia iprogram iURL ias ia iguidance ito ichange ithe iview.

## Chapter5

**Workingof TrackIt App**

AswearemovingtowardstheworkingoftheTrackItapp,letstakeabrieflookonthetechnologiesused.

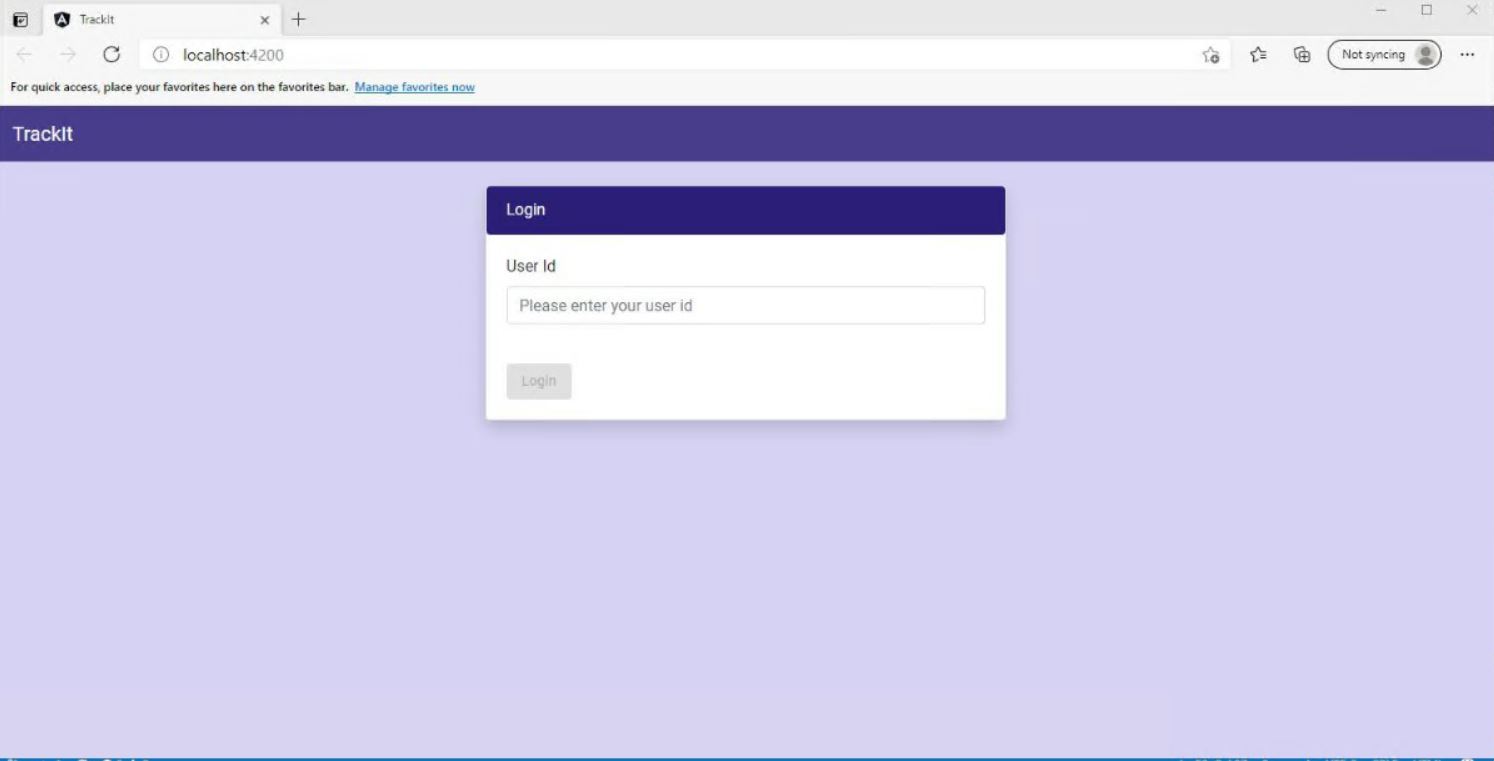
* Javaforbackendinspring.
* HTMLalongwithCSSandBootstrapfordesigningandClientenddevelopment.
* TypescriptforAngulardesigning.
* MongoDB Databaseforloginandregisteringuser.

Nowletsbeginwiththeworkingoftheapplication.

Tostarttheappweneedtofirststartthespringapplicationdevelopedineclipse.ThisAPIcontainsanentityclasscalleduserinitforstoringthedetailsoftheuserinthedatabase.

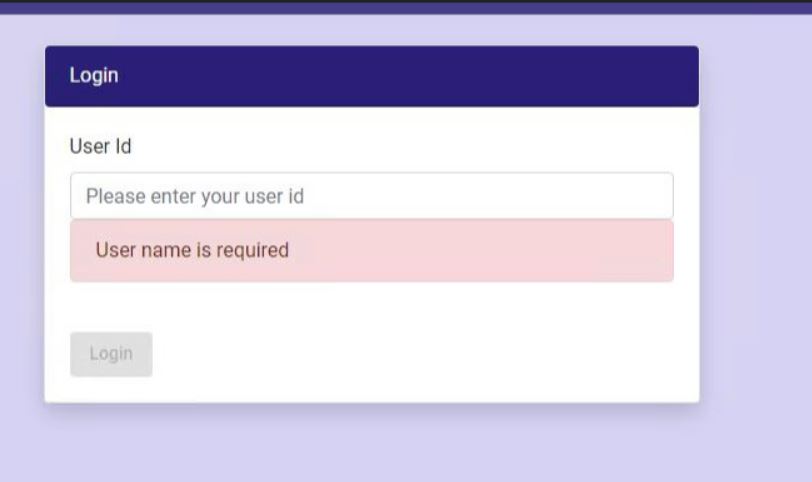
NowafterstartingthespringAPIwethenneedtoconnectourMongoDBdatabase.ForconnectingMongoDB databasewejusthavetogotothelocalhostportonwhichthedatabaseisrunningandpresstheconnectbutton.

Whenbothofthesethingsaredonethenweneedtostartourangularapplicationdevelopedforthefrontendofourapplication.Bydefaultangularrunsonportnumber4200.

**Fig5.1LoginPage**

Track It application is based on authorization based login rather than authentication based login authorization here is for trainee and batch-owner. In the Input field User Id of the batch-owner or the trainee is given and after clicking on the login button it takes us to the respective batch-owner/Trainee page.

If User-id is not mentioned in employee database then an alert box is shown displaying alert messages. Employee database consist of user-id and respective roles for the id. On Back-end call to employee database is made to get user role then on the bases of received role user is redirected to respective pages.



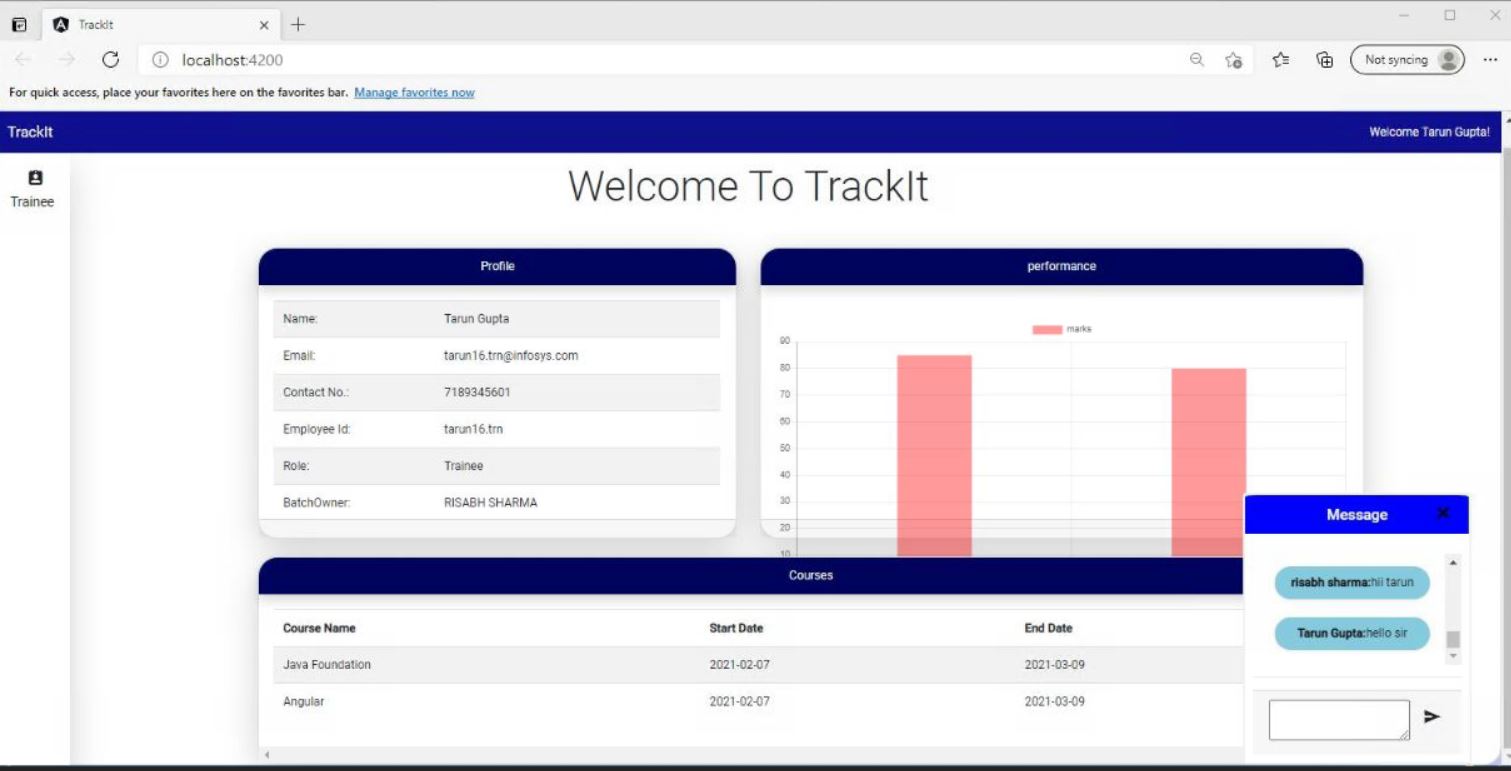
**Fig5.2Validationsinloginpage**

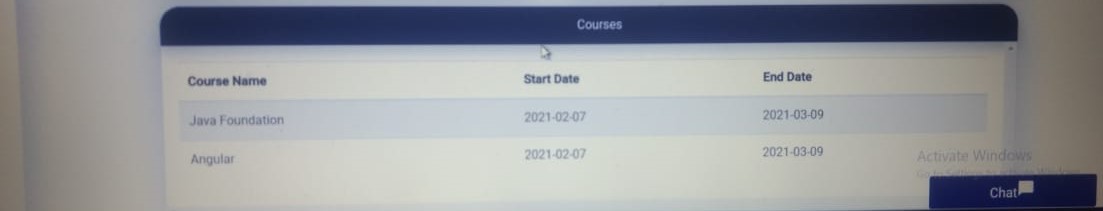
Fig5.2showsthevalidationsthataregiventothisform.Properregexpatternsaregiventoboththefields.Emailidmustbeinaformatof“[*example@xyz.com*](mailto:example@xyz.com)*”*andpasswordmustbeminimum8characterslongandmustcontain1digitand1specialcharacter.

### Trainee DashBoard

When the user enters their user id the employee api gives back data to the frontend along with their roles ,If the employee having Trainee as a role in database It is routed to trainee dashboard using routing in angular.

Trainee dashboard consist of various cards named as profile, performance, courses and also a chat pop up to communicate with batchowner.

**Fig5.3.Trainee Dashboard**

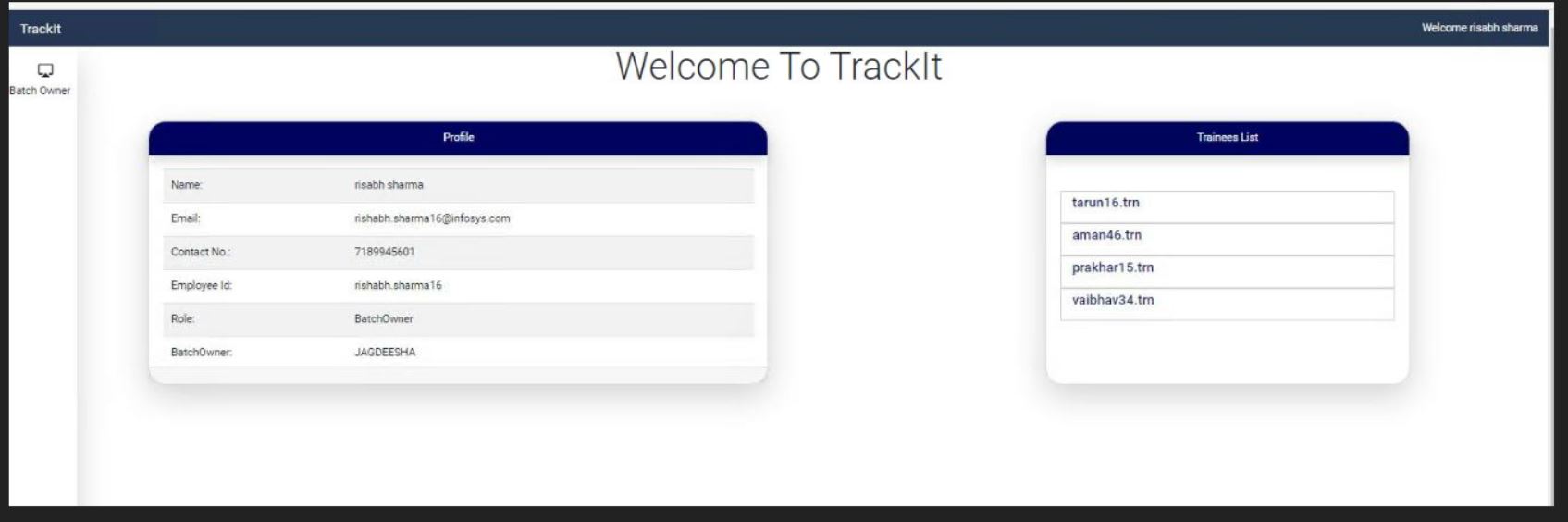


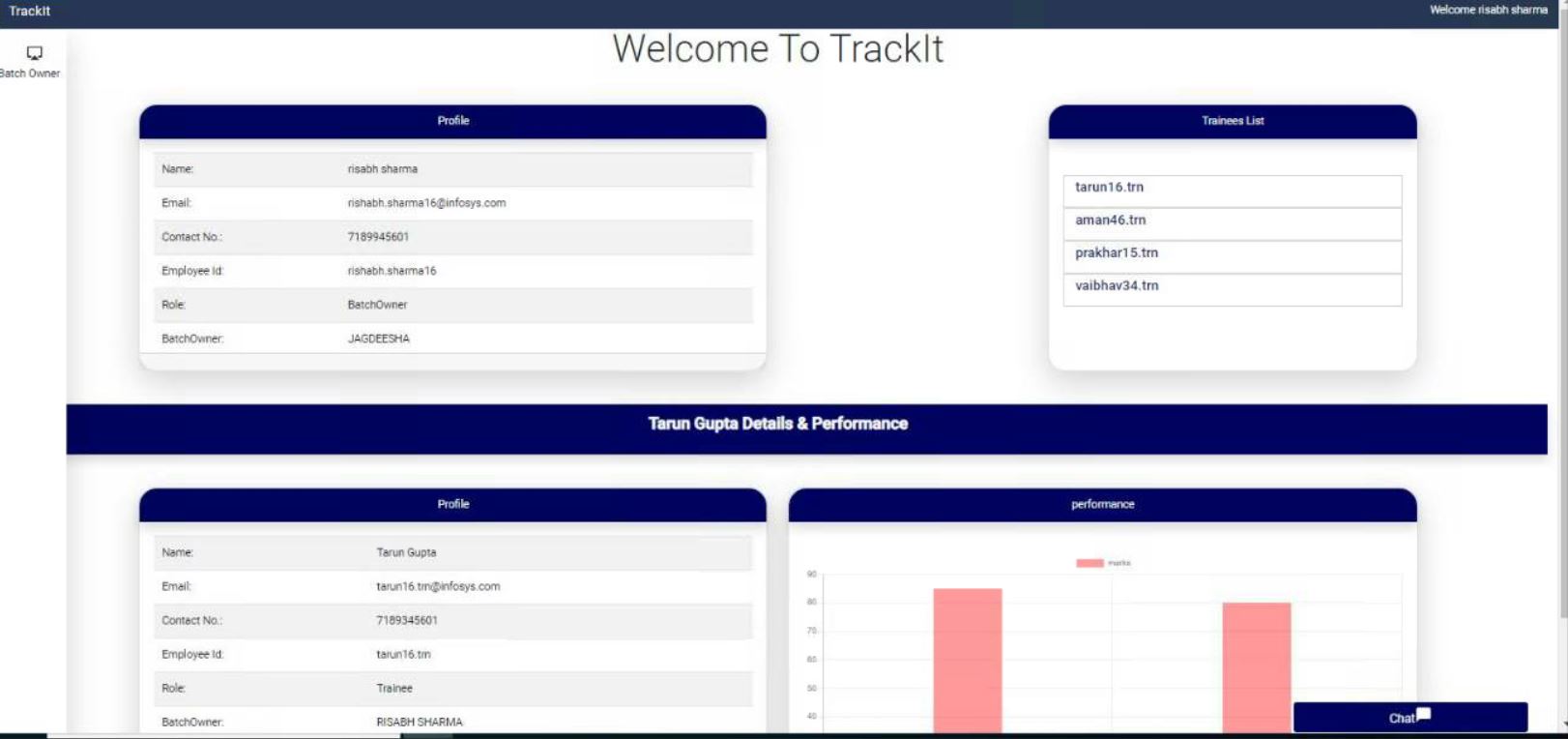
**Fig5.4 Trainee Dashboard**

### BatchOwner

When the user enters their user id the employee api gives back data to the frontend along with their roles ,If the employee having BatchOwner as a role in database It is routed to BatchOwner dashboard using routing in angular.

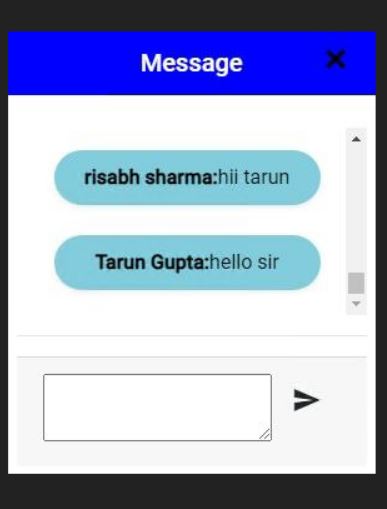
BatchOwner dashboard consist of various cards named as profile, performance, courses and also a chat pop up to communicate with Trainee.Trainee Details are also accessible to BatchOwner on dynamic basis.

**Fig5.5 Batch-Owner Dashboard**

**Fig5.6 Batch-Owner Dashboard**

### Chat-box

It is a communication network established between batch owner and trainee. This is so that queries put up by trainee can be resolved by Batch-owner.

****

**Fig5.7 Pop up Chat-box**

## CONCLUSION

* Theprojecthasvariousrealtimeongoingapplicationsforconducting training, Facilitating virtual on-boarding etc.
* This project can help batch-owner to monitor trainee performance and data on real time basis and guide them.
* This project can also be extended by the client and add their own cards as per their needs
* Modular nature of this project comes with implementation of micro services which enhances availability and up time of project.

## REFERENCES

1. "SpringBootTutorials|JavaDevelopmentJournal",*JavaDevelopmentJournal*,2021.[Online].Available:https://[www.javadevjournal.com/spring-](http://www.javadevjournal.com/spring-)boot/.[Accessed:18-May-2021].
2. V.Balasubramaniam,"DefiningJPAEntities|Baeldung",*Baeldung*,2021.[Online].Available:https://[www.baeldung.com/jpa-entities.](http://www.baeldung.com/jpa-entities)[Accessed:18-May-2021].
3. J.Journal,"SpringBootAnnotations",*JavaDevelopmentJournal*,2021.[Online].Available:https://[www.javadevjournal.com/spring-boot/spring-](http://www.javadevjournal.com/spring-boot/spring-)boot-annotations/.[Accessed:18-May-2021].
4. "TheiSpringi@Controlleriandi@RestControlleriAnnotationsi|Baeldung", *Baeldung*, 2021. [Online]. Available:https://[www.baeldung.com/spring-controller-vs-restcontroller.](http://www.baeldung.com/spring-controller-vs-restcontroller)[Accessed:18-May-2021].
5. "Angular", *Angular.io*, 2021. [Online]. Available:https://angular.io/guide/architecture-components.[Accessed:18-May-2021].
6. "Angular", *Angular.io*, 2021. [Online]. Available:https://angular.io/api/common/NgIf.[Accessed:18-May-2021].
7. "\*ngForDirectiveinAngular|DigitalOcean",*DigitalOcean*,2021.[Online].Available: https://[www.digitalocean.com/community/tutorials/angular-](http://www.digitalocean.com/community/tutorials/angular-)ngfor-directive.[Accessed:18-May-2021].
8. "Angular", *Angular.io*, 2021. [Online]. Available:https://angular.io/api/common/NgSwitch.[Accessed:18-May-2021].