PHP STUDY

```
<?php     //Connection file
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Credentials: true");
header("Access-Control-Max-Age: 1000");
header("Access-Control-Allow-Headers: X-Requested-With, Content-Type, Origin, Cache-Control, Pragma, Authorization, Accept, Accept-Encoding");
header("Access-Control-Allow-Methods: PUT, POST, GET, OPTIONS, DELETE");

$title="Mess mgt";

$dbhost="localhost";
$dbuser="root";
$dbpassword="";
$dbname="mymess";

$conn = new mysqli($dbhost,$dbuser,$dbpassword,$dbname);
?>
```

```
<?php     //Login file

include("include-files/connection.php");
require_once 'jwtencode.php';

header('Content-type: application/json');
```

```
$postdata = file_get_contents("php://input");

$postdata2=json_decode($postdata,true);



if($_SERVER['REQUEST_METHOD'] == "POST" && isset($postdata2['loginid'])
&& isset($postdata2['password']) )

{


$loginid = $postdata2['loginid'];

$password = $postdata2['password'];

$getmac = execnil(9673);





        $sql = "SELECT * FROM authentic_user WHERE loginid = '$loginid' and
password = '$password'";

        $result =$conn->query($sql)or trigger_error(mysql_error());


        $row =mysqli_fetch_array($result);

        $user_role=$row['role'];

        $count = mysqli_num_rows($result);

        $data[]=$postdata2;

        if($count == 1)

        {

                //echo "count1";

                if($user_role==0)
        {
            $data2 = array("status" => false, 
"responce_status_massage" => "Wrong role"  );
            echo json_encode($data2,JSON_NUMERIC_CHECK);
        }
        else
        {
            $mess_config = "SELECT * FROM mess_config WHERE 
system_key = '$getmac'";
            $resultconfig =$conn->query($mess_config)or 
trigger_error(mysql_error());
            $rowconfig =mysqli_fetch_array($resultconfig);
            $countconfig = mysqli_num_rows($resultconfig);
            if($countconfig == 1)
            {



                $sqtime="SELECT * FROM `mess_config`";
                $sql_result_time = $conn->query($sqtime);
                $row = $sql_result_time->fetch_assoc();






                $configdata[]=$row;



    ///*********************************************************
***************************
```

```
                                        //Auto operation during login process. Bill Generation
monthly here ***////


                                        $data2 = array("status" => true,
"responce_status_massage" => "Data Found" ,"token" => $jwt , "data"=> $data
, "configdata"=> $configdata);

                                        echo json_encode($data2,JSON_NUMERIC_CHECK);

                                        /* header("location:../cronjobs/cron-job-check.php");
*/


                                        ////
****************************************************************
******************

                                        //This code will generate croon operation output and
then redirect to home page.

                              }

                              else

                              {

                                        $data2 = array("status" => false,
"responce_status_massage" => "Non activated software1"  , "data"=> $data);

                                        echo json_encode($data2,JSON_NUMERIC_CHECK);

                                        /* header("location:../register.php?msg=कृपया
सॉफ्टवेर आक्टीवेट करण्यासाठी NP Solutions ला संपर्क साधा"); */

                              }

                    }


          }

          else

          {
```

```
        $mess_config = "SELECT * FROM mess_config WHERE system_key = '$getmac'";
        $resultconfig =$conn->query($mess_config)or trigger_error(mysql_error());
        $rowconfig =mysqli_fetch_array($resultconfig);
        $countconfig = mysqli_num_rows($resultconfig);
        if($countconfig == 1)
        {
            $data2 = array("status" => false, "responce_status_massage" => "Invalid Id Password"  , "data"=> $data);
            echo json_encode($data2,JSON_NUMERIC_CHECK);
            /* header("location: ../index.php?msg=पासवर्ड किंवा आयडी चुकले आहे"); */
        }
        else
        {
            $data2 = array("status" => false, "responce_status_massage" => "Non activated software"  , "data"=> $data);
            echo json_encode($data2,JSON_NUMERIC_CHECK);
            /* header("location:../register.php?msg=कृपया सॉफ्टवेर आक्टीवेट करण्यासाठी NP Solutions ला संपर्क साधा"); */
        }



    }



}
```

```
else

{


        $data = array("status" => false, "responce_status_massage" => "Invalid method");

        echo json_encode($data);

}

$conn->close();

?>
```

Student project

Insert data in student table

```
INSERT INTO `student`(`student_name`, `roll_no`, `class`, `city`) VALUES ('vivek',11,5,'Jalgaon')
```

Delete data in student table

```
"DELETE FROM `student` WHERE `student`.`student_id` = 2"
```

Update data in student table

```
update student set roll_no=15 where student_name like 'kunal'
```

Fetching all students who have selected Marathi subject

```
select student.student_name,subject.subject_name from student join stud_subject on (student.student_id=stud_subject.student_id) join subject on (stud_subject.subject_id=subject.subject_id) WHERE subject.subject_name='Marathi'
```

Fetching data using join in descending order

```
select student.student_name,subject.subject_name,student_marks.marks
from student join student_marks on
(student.student_id=student_marks.student_id) join subject on
(student_marks.subject_id=subject.subject_id) order by student_marks.marks
desc
```

Fetching data using join and group by

```
select subject.subject_name,count(stud_subject.student_id) from subject join
stud_subject on subject.subject_id=stud_subject.subject_id GROUP BY
subject.subject_id
```

Angular IMP

Data.service.ts

```
import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';


@Injectable({

  providedIn: 'root'

})

export class DataService {


  API_URL  =  'http://localhost/mess-api';

  constructor(private http: HttpClient) { }
```

```
//1
postlogin($formdata)
{
  return this.http.post(`${this.API_URL}/login.php`,$formdata);
}
//2
loggedin()
{
  return !!localStorage.getItem('token')
}
//3
gettoken()
{
  return localStorage.getItem('token')
}
//3
listofstudents()
{
  return this.http.post(`${this.API_URL}/student_list.php`,123);
}
//4
noofstudentactive()
{
  return this.http.post(`${this.API_URL}/no-of-students-active.php`,123);
}
```

```
}
```

Auth.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, Router } from
'@angular/router';
/*import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot } from
'@angular/router';
import { Observable } from 'rxjs';*/
import { DataService } from './data.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(private data: DataService,private _rout: Router){}

  canActivate(): boolean
  {
    if(this.data.loggedin())
    {
      return true
    }
    else
    {
      this._rout.navigate(['/login'])
      return false
    }

  }
  /*canActivate(
    next: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean> | Promise<boolean> | boolean {
    return true;
  }*/
}
```

Interceptor.service.ts

```
import { Injectable,Injector } from '@angular/core';
import { DataService } from './data.service';
import { HttpInterceptor } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class TokenInterceptorService implements HttpInterceptor {

  constructor(private injector: Injector) { }
  intercept(req,next)
  {
    let authoriser = this.injector.get(DataService)
    let tokenizedreq = req.clone({
       setHeaders:{
        Authorization: `Bearer ${authoriser.gettoken()}`
       }
    })
    return next.handle(tokenizedreq)
  }

}
```

App-routing.modules.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { AdminComponent } from './layouts/admin/admin.component';
import { DashboardComponent } from './dashboard/dashboard.component';
import { PagesComponent } from './layouts/pages/pages.component';
import { LoginComponent } from './login/login.component';
import { AuthGuard } from './auth.guard';

const routes: Routes = [
  {
    path:'',  redirectTo: 'dashboard',  pathMatch: 'full'
  },
  {
    path:'',
    component: AdminComponent,
    canActivate: [AuthGuard],
    children:[
      {
        path:'dashboard',
        component:  DashboardComponent
      }
    ]
  },
  {
    path:'',
    component: PagesComponent,
    children:[
      {
        path:'login',
        component: LoginComponent
      }
    ]
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Agnular login form

```
<form  (ngSubmit)="loginform()"   >

        <div class="form-group">
          <label class="label">Username</label>
          <div class="input-group">
            <input type="text"  [(ngModel)]="userlogin.loginid" class="form-control" name="loginid"  placeholder="Username" required >
            <div class="input-group-append">
              <span class="input-group-text">
                <i class="mdi mdi-check-circle-outline"></i>
              </span>
            </div>
          </div>
        </div>
        <div class="form-group">
          <label class="label">Password</label>
          <div class="input-group">
            <input type="password" [(ngModel)]="userlogin.password" name="password"  class="form-control" placeholder="*********"  required>
            <div class="input-group-append">
              <span class="input-group-text">
                <i class="mdi mdi-check-circle-outline"></i>
```

```
      </span>
    </div>
  </div>
</div>
<div class="form-group">
    <input  type="submit"  class="btn btn-primary submit-btn btn-block" value="submit"/>
</div>
<div class="form-group d-flex justify-content-between">
  <div class="form-check form-check-flat mt-0">
    <label class="form-check-label">
      <input type="checkbox" class="form-check-input" checked> Keep me signed in
    </label>
  </div>
  <a href="#" class="text-small forgot-password text-black">Forgot Password</a>
</div>
<div class="form-group">
  <button class="btn btn-block g-login">
    <img class="mr-3" src="" alt="">Log in with Google</button>
</div>
<div class="text-block text-center my-3">
  <span class="text-small font-weight-semibold">Not a member ?</span>
  <a href="register.html" class="text-black text-small">Create new account</a>
</div>
```

```
        </form>
```

Login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Form } from '@angular/forms';
import { DataService } from '../data.service';
import { Router } from '@angular/router';
import { Observable } from 'rxjs';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent implements OnInit {

  public logobj: object;
  constructor(private data: DataService,private _rout: Router) { }

  ngOnInit() {
  }
  userlogin = {}
  loginform() {
    console.log(this.userlogin);
      return this.data.postlogin(this.userlogin).subscribe(data => {
        this.logobj = data
        console.log(data);
        if(this.logobj['status'])
        {
          localStorage.setItem('token',<any>data['token'])
          this._rout.navigate(['/dashboard'])

        }
        else
        {
          localStorage.clear();
          console.log('No token');
        }
    })
  }

}
```

App.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { DashboardComponent } from './dashboard/dashboard.component';
import { LoginComponent } from './login/login.component';
import { AdminComponent } from './layouts/admin/admin.component';
import { PagesComponent } from './layouts/pages/pages.component';
import { FooterComponent } from './footer/footer.component';
import { SidebarComponent } from './sidebar/sidebar.component';
import { NavbarComponent } from './navbar/navbar.component';
import { HttpClientModule, HTTP_INTERCEPTORS } from '@angular/common/http';
import { FormsModule } from '@angular/forms';
import { DataService } from './data.service';
import { AuthGuard } from './auth.guard';
import { TokenInterceptorService } from './token-interceptor.service';

@NgModule({
  declarations: [
    AppComponent,
    DashboardComponent,
    LoginComponent,
    AdminComponent,
    PagesComponent,
    FooterComponent,
    SidebarComponent,
    NavbarComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [DataService,AuthGuard,
    {
      provide:HTTP_INTERCEPTORS,
      useClass: TokenInterceptorService,
      multi:true
    }],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

//Dashboard.component.ts check login and redirect

```
import { Component, OnInit } from '@angular/core';
import { DataService } from '../data.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.scss']
})
export class DashboardComponent implements OnInit {
  homedata = {}
  noofstuds = {}
```

```
constructor(private data: DataService,private _rout: Router) { }


ngOnInit() {
  console.log(this.data.loggedin());
  this.loadhome();
  this.noofstdhome();


}
loadhome()
{
  return this.data.listofstudents().subscribe(data => {
    this.homedata = data
    console.log(this.homedata)
    if(!this.homedata)
    {
      if(this.homedata['responce_status_massage']=='Invalid Token')
      {
          localStorage.clear();
          this._rout.navigate(['/login'])
      }
      else
      {
        alert(this.homedata['responce_status_massage']);
      }
    }
    else { }
  })
}

noofstdhome()
{
  return this.data.noofstudentactive().subscribe(data => {
    this.noofstuds = data
    console.log(this.noofstuds)
    if(!this.noofstuds)
    {
      if(this.noofstuds['responce_status_massage']=='Invalid Token')
      {
          localStorage.clear();
          this._rout.navigate(['/login'])
      }
      else
      {
        alert(this.noofstuds['responce_status_massage']);
      }
    }
    else { }

  })
}
```

```
}
```