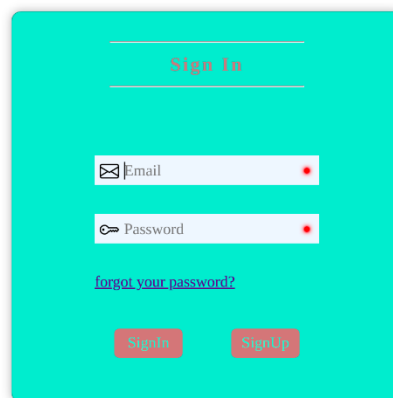
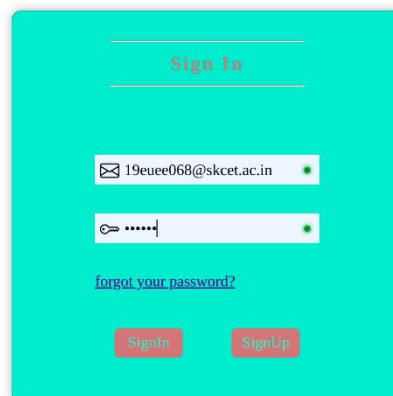


Cloud Storage Using Servlet

Front End:

Login page:


 Screenshots:

A login form titled 'Sign In' with a red underline. It contains two input fields: 'Email' with an envelope icon and 'Password' with a key icon. Both fields have a red asterisk on the right. Below the fields is a blue link 'forgot your password?'. At the bottom are two red buttons: 'SignIn' and 'SignUp'.A login form titled 'Sign In' with a red underline. The 'Email' field is filled with '19euee068@skcet.ac.in' and has a green checkmark on the right. The 'Password' field is filled with '.....' and has a green checkmark on the right. Below the fields is a blue link 'forgot your password?'. At the bottom are two red buttons: 'SignIn' and 'SignUp'.

localhost:7060/forgotpass.jsp




Sign Up





Sign Up



Code:

Login.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="jcloud.ServerPage" %>

<%
    if(ServerPage.auth(request)) response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <link rel="stylesheet" href="css/signin.css">
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boot
strap-icons@1.7.1/font/bootstrap-icons.css">

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.
0/jquery.min.js"></script>
        <script src="js/signin.js"></script>

        <title>SignIn</title>
    </head>
    <body>
        <div id="signin">
            <div id="title"><hr>Sign In<hr></div>
            <form method="post" name="signin" action="/">
                <div id="email_in"><i class="bi bi-
envelope"></i><input type="text" placeholder="Email">
                    <div id="email_v"></div>
                </div>
                <div id="password_in"><i class="bi bi-
key"></i><input type="password" placeholder="Password">
                    <div id="password_v"></div>
                </div>
                <a href="forgotpass.jsp" style="margin-
left: 85px;">forgot your password?</a>
            </form>
            <div id="button_cont">
                <div id="signin_button" onclick="sign_in();">SignIn</d
iv>
```

```

        <div id="signup_button" onclick="swap(document.getElementBy
entById('signin'));">SignUp</div>
    </div>
</div>
<div id="signup">
    <div id="title"><hr>Sign Up<hr></div>
    <div id="profile_pic">
        
        <div id="layer" onclick="upload_img();"></div>
    </div>
    <div id="profile_pic_v"></div>
    <form method="post" name="signup" action="/">
        <div id="fullname"><i class="bi bi-
person"></i><input type="text" placeholder="Fullname">
            <div id="fullname_v"></div>
        </div>
        <div id="email"><i class="bi bi-
envelope"></i><input type="email" placeholder="Email">
            <div id="email_v2"> </div>
        </div>
        <div id="password"><i class="bi bi-
key"></i><input type="password" placeholder="password">
            <div id="password_v2"></div>
        </div>
    </form>
    <div id="button_cont">
        <div id="signin_button" onclick="swap(document.getElementBy
entById('signup'));">SignIn</div>
        <div id="signup_button" onclick="sign_up();">SignUp</d
iv>
    </div>
</div>
<script>

</script>
</body>
</html>

```

Login.css:

```
:root{
  --estatus:red;
  --pstatus:red;
  --e2status: red;
  --p2status: red;
  --fnstatus: red;
  --ppstatus: red;
}

body{
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

input{
  all: unset;
}

#signin{
  position: absolute;
  width: 400px;
  height: 400px;
  border: .2px solid rgba(0, 0, 0, 0.377);
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.288);
  transition: opacity 1s;
  background-color: rgb(37, 226, 201);
  z-index: 1;
}

#signin #title{
  width: 200px;
  margin: 30px auto;
  text-align: center;
  color: rgb(189, 129, 129);
  font-weight: bold;
  font-size: 20px;
  letter-spacing: 2px;
}
```

```
#email_in{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 5px;
    margin: 10px auto;
    margin-top: 70px;
    position : relative;
}
```

```
#password_in{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 5px;
    margin: 30px auto;
    position : relative;
}
```

```
#password_in i{
    margin-right: 5px;
    font-size: 20px;
}
```

```
#email_in i{
    margin-right: 5px;
    font-size: 20px;
}
```

```
#email_in input{
    position :relative;
    top: -3px;
}
```

```
#password_in input{
    position :relative;
    top: -3px;
}
```

```
#email_v{
    width :5px;
    height: 5px;
    border-radius: 50%;
}
```

```
background-color: var(--estatus);
box-shadow: 0 0 1px var(--estatus),0 0 2px var(--estatus),0 0 4px var(--
estatus),0 0 6px var(--estatus);
position: absolute;
top: 13px;
right: 10px;
}

#password_v{
width :5px;
height: 5px;
border-radius: 50%;
background-color: var(--pstatus);
box-shadow: 0 0 1px var(--pstatus),0 0 2px var(--pstatus),0 0 4px var(--
pstatus),0 0 6px var(--pstatus);
position: absolute;
top: 13px;
right: 10px;
}

#email_v2{
width :5px;
height: 5px;
border-radius: 50%;
background-color: var(--e2status);
box-shadow: 0 0 1px var(--e2status),0 0 2px var(--e2status),0 0 4px var(--
e2status),0 0 6px var(--e2status);
position: absolute;
top: 13px;
right: 10px;
}

#password_v2{
width :5px;
height: 5px;
border-radius: 50%;
background-color: var(--p2status);
box-shadow: 0 0 1px var(--p2status),0 0 2px var(--p2status),0 0 4px var(--
p2status),0 0 6px var(--p2status);
position: absolute;
top: 13px;
right: 10px;
}

#fullname_v{
```

```

    width :5px;
    height: 5px;
    border-radius: 50%;
    background-color: var(--fnstatus);
    box-shadow: 0 0 1px var(--fnstatus),0 0 2px var(--fnstatus),0 0 4px var(--
fnstatus),0 0 6px var(--fnstatus);
    position: absolute;
    top: 13px;
    right: 10px;
}

#signup{
    position: absolute;
    width: 400px;
    height: 400px;
    border: .2px solid rgba(0, 0, 0, 0.377);
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.288);
    transition: opacity 1s;
    background-color: blanchedalmond;
    z-index: 0;
}

#signup #title{
    width: 200px;
    margin: 30px auto;
    text-align: center;
    color: rgb(129, 143, 189);
    font-weight: bold;
    font-size: 20px;
    letter-spacing: 2px;
    margin-bottom: 0;
}

#profile_pic{
    border-radius: 50%;
    width: 60px;
    height: 60px;
    margin: 15px auto;
    margin-top: 10px;
    overflow: hidden;
    position: relative;
    border: .2px solid rgba(129, 143, 189, 0.5);
    background-color: rgb(129, 143, 189);
}

```



```
#profile_pic img{
    width: 60px;
    height: 60px;
}

#profile_pic_v{
    width :5px;
    height: 5px;
    border-radius: 50%;
    background-color: var(--ppstatus);
    box-shadow: 0 0 1px var(--ppstatus),0 0 2px var(--ppstatus),0 0 4px var(--
ppstatus),0 0 6px var(--ppstatus);
    position: absolute;
    top: 115px;
    left: 300px;
}

#profile_pic #layer{
    width: 100%;
    height: 38%;
    background-color: rgba(0, 0, 0, 0.4);
    position: absolute;
    bottom: 0;
    backdrop-filter: blur(1px);
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}

#profile_pic #layer img{
    width: 23px;
    height: 23px;
    filter: invert(1);
}

#fullname{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 6px;
    padding-left: 25px;
    margin: 10px auto;
```

```
    margin-top: 20px;
    position: relative;
}

#email{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 6px;
    padding-left: 27px;
    margin: 20px auto;
    position: relative;
}

#password{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 27px;
    margin: 20px auto;
    position: relative;
}

#password i{
    position: absolute;
    left: 3px;
    margin-right: 5px;
    font-size: 20px;
}

#email i{
    position: absolute;
    top: 5.5px;
    left: 4.5px;
    margin-right: 5px;
    font-size: 18px;
}

#fullname i{
    position: absolute;
    top: 4px;
    left: 3.5px;
    font-size: 20px;
}
```

```
}

#button_cont{
    width: 50%;
    position: absolute;
    bottom: 40px;
    left: 25%;
}

#signup #button_cont div{
    background-color: rgb(129, 143, 189);
    color: rgb(255, 244, 228);
}

#button_cont #signin_button{
    float: left;
}

#button_cont #signup_button{
    float: right;
}

#button_cont div{
    width: 70px;
    height: 30px;
    color: aquamarine;
    background-color: rgb(197, 125, 125);
    margin: 5px;
    padding: 5px;
    box-sizing: border-box;
    text-align: center;
    border-radius: 5px;
    cursor: pointer;
}

#button_cont div:hover{
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);
}
```

Login.js:

```
var file_in;

function swap(self) {
  if(self.id=='signup'){
    self.style.opacity="0"
    document.getElementById("signin").style.opacity="1"
    setTimeout(()=>{
      document.getElementById("signin").style.zIndex=1
      self.style.zIndex=0
    },1000)
  }
  else{
    self.style.opacity="0"
    document.getElementById("signup").style.opacity="1"
    setTimeout(()=>{
      document.getElementById("signup").style.zIndex=1
      self.style.zIndex=0
    },1000)
  }
}

function upload_img(){
  file_in = document.createElement('input')
  file_in.type='file'
  file_in.accept='image/*'
  file_in.click()
  file_in.addEventListener('change',function(ev){
    console.log(this.files)
    if (this.files && this.files[0]) {
      var reader = new FileReader();
      reader.onload = function (e) {
        $("#profile_pic_img").attr('src', e.target.result);
        document.querySelector(":root").style.setProperty("--
ppstatus","green");
      }
      reader.readAsDataURL(this.files[0]);
    }
    else{
      document.querySelector(":root").style.setProperty("--
ppstatus","red");
    }
  })
}
```

```

function sign_in() {
    var e=$("#email_in input"),p=$("#password_in input")
    if(ValidateEmail(e.val()) && p.val().length>0){
        $.ajax({
            type: "POST",
            url: "ServerPage",
            data: {
                type: "SIGNIN",
                email: e.val(),
                password: p.val()
            },
            success: (json)=>{
                console.log(json)
                if(json.status=="SIGNIN SUCCESS") {
                    window.location.href='index.jsp';
                }
            }
        })
    }
}

function sign_up(){
    var f=$("#fullname input"), e=$("#email input"),p=$("#password input")
    if(ValidateEmail(e.val()) && p.val().length>0 && f.val().length>0 && file_in.
files.length>0){
        var d=new FormData();
        d.append("img",file_in.files[0])
        d.append("fullname",f.val())
        d.append("type","SIGNUP")
        d.append("email",e.val())
        d.append("password",p.val())
        $.ajax({
            type: "POST",
            url: "ServerPage",
            dataType: "text/json",
            processData: false,
            contentType: false,
            data: d,
            success: (json)=>{
                console.log(json)
                if(json.status=="SIGNUP SUCCESS") {
                    window.location.href='index.jsp';
                }
            },
        })
    }
}

```

```

        error:(e)=>{
            window.location.href='index.jsp';
        }
    })
}
}
}
document.addEventListener("DOMContentLoaded",()=>{
    $("#email_in input")[0].addEventListener("input",()=>{
        var email=$("#email_in input").val();
        if(ValidateEmail(email)){
            document.querySelector(":root").style.setProperty("--
estatus","green");
        }else{
            document.querySelector(":root").style.setProperty("--estatus","red");
        }
    })
    $("#password_in input")[0].addEventListener("input",()=>{
        var pass=$("#password_in input").val();
        if(pass.length){
            document.querySelector(":root").style.setProperty("--
pstatus","green");
        }else{
            document.querySelector(":root").style.setProperty("--pstatus","red");
        }
    })
    $("#email input")[0].addEventListener("input",()=>{
        var email=$("#email input").val();
        if(ValidateEmail(email)){
            document.querySelector(":root").style.setProperty("--
e2status","green");
        }else{
            document.querySelector(":root").style.setProperty("--
e2status","red");
        }
    })
    $("#password input")[0].addEventListener("input",()=>{
        var pass=$("#password input").val();
        if(pass.length){
            document.querySelector(":root").style.setProperty("--
p2status","green");
        }else{

```

```

        document.querySelector(":root").style.setProperty("--
p2status", "red");
    }

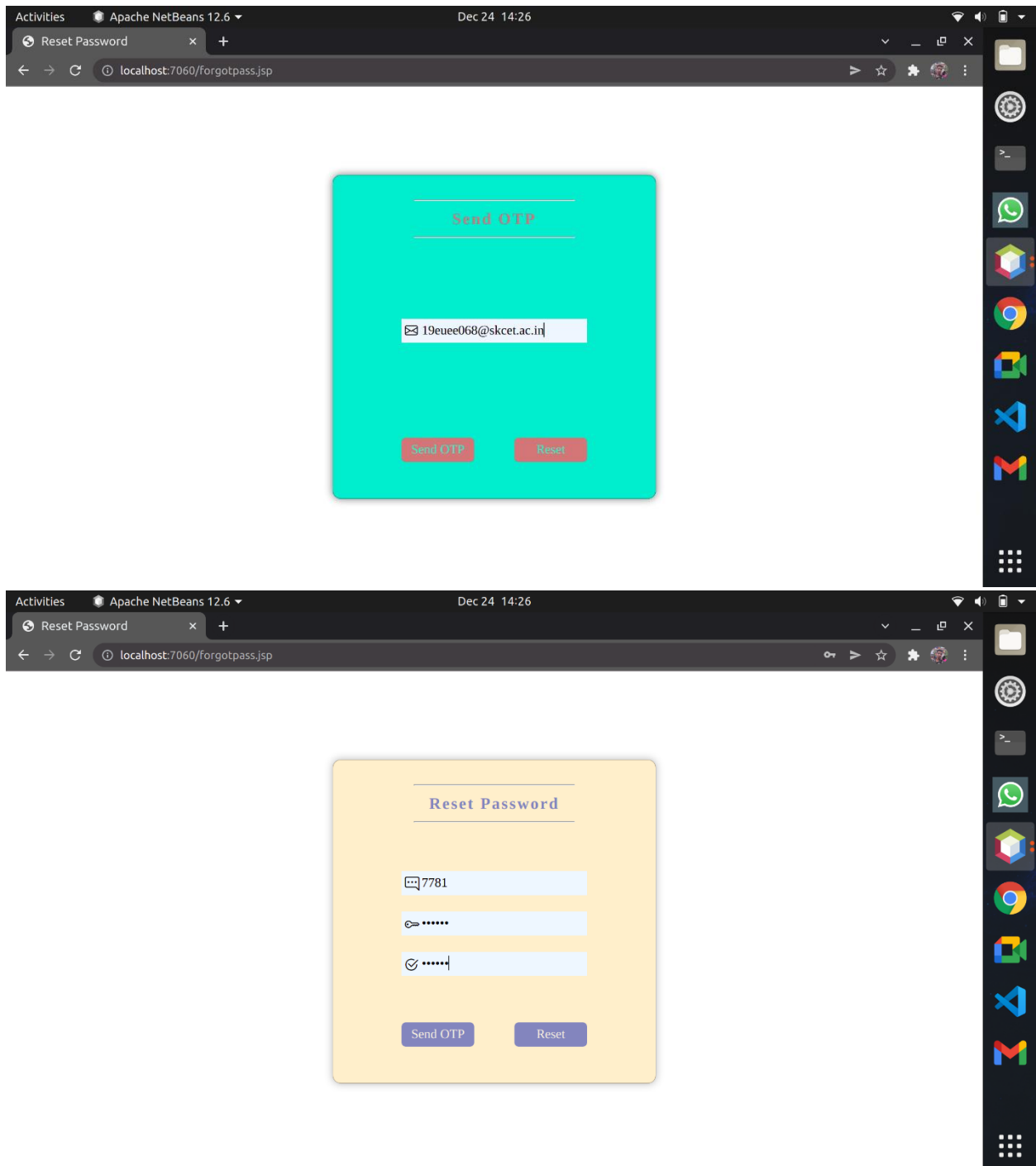
    })
    $("#fullname input")[0].addEventListener("input", ()=>{
        var fn=$("#fullname input").val();
        if(fn.length){
            document.querySelector(":root").style.setProperty("--
fnstatus", "green");
        }else{
            document.querySelector(":root").style.setProperty("--
fnstatus", "red");
        }
    })
})
function ValidateEmail(mail)
{
    if (/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/ .test(mail))
    {
        return (true)
    }

    return (false)
}

```

Forgot Password Page:

Screenshots:



Forgotpass.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="jcloud.ServerPage" %>

<%
    if(ServerPage.auth(request)) response.sendRedirect("index.jsp");
%>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Reset Password</title>
    <link rel="stylesheet" href="css/resetpass.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.1/font/bootstrap-icons.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jq
uery.min.js"></script>
    <script src="js/resetpass.js"></script>
</head>
<body>
    <div id="send_otp">
        <div id="title"><hr>Send OTP<hr></div>
        <form method="post" name="signin" action="/">
            <div id="email_in"><i class="bi bi-
envelope"></i><input type="text" placeholder="Email">
            </div>
        </form>
        <div id="button_cont">
            <div id="send_otp_button" onclick="sendOtp();">Send OTP</d
iv>
            <div id="reset_password_button" onclick="swap(document.get
ElementById('send_otp'));">Reset</div>
        </div>
    </div>
    <div id="reset_password">
        <div id="title"><hr>Reset Password<hr></div>

        <form method="post" name="signup" action="/">
            <div id="fullname"><i class="bi bi-chat-right-
dots"></i><input type="text" placeholder="OTP"></div>
```

```

        <div id="email"><i class="bi bi-key"></i><input type="password" placeholder="New Password"></div>
        <div id="password"><i class="bi bi-check2-circle"></i><input type="password" placeholder="Re-type Password"></div>
    </form>
    <div id="button_cont">
        <div id="send_otp_button" onclick="swap(document.getElementById('reset_password'));">Send OTP</div>
        <div id="reset_password_button" onclick="reset();">Reset</div>
    </div>
</div>
</body>
</html>

```

Resetpass.css:

```

body{
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}

input{
    all: unset;
}

#send_otp{
    position: absolute;
    width: 400px;
    height: 400px;
    border: .2px solid rgba(0, 0, 0, 0.377);
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.288);
    transition: opacity 1s;
    background-color: rgb(37, 226, 201);
    z-index: 1;
}

#send_otp #title{

```

```
width: 200px;
margin: 30px auto;
text-align: center;
color: rgb(189, 129, 129);
font-weight: bold;
font-size: 20px;
letter-spacing: 2px;
}

#email_in{
  box-sizing: border-box;
  background-color: aliceblue;
  width: 230px;
  height: 30px;
  padding-top: 5px;
  padding-left: 25px;
  margin: 10px auto;
  margin-top: 100px;
  position: relative;
}

#password_in{
  box-sizing: border-box;
  background-color: aliceblue;
  width: 230px;
  height: 30px;
  padding-top: 5px;
  padding-left: 25px;
  margin: 30px auto;
}

#reset_password{
  position: absolute;
  width: 400px;
  height: 400px;
  border: .2px solid rgba(0, 0, 0, 0.377);
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.288);
  transition: opacity 1s;
  background-color: blanchedalmond;
  z-index: 0;
}

#reset_password #title{
  width: 200px;
```

```

    margin: 30px auto;
    text-align: center;
    color: rgb(129, 143, 189);
    font-weight: bold;
    font-size: 20px;
    letter-spacing: 2px;
    margin-bottom: 0;
}

#profile_pic{
    border-radius: 50%;
    width: 60px;
    height: 60px;
    margin: 15px auto;
    margin-top: 10px;
    overflow: hidden;
    position: relative;
    border: .2px solid rgba(0, 0, 0, 0.4);
}

#profile_pic img{
    width: 60px;
    height: 60px;
}

#profile_pic #layer{
    width: 100%;
    height: 38%;
    background-color: rgba(0, 0, 0, 0.4);
    position: absolute;
    bottom: 0;
    backdrop-filter: blur(1px);
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}

#profile_pic #layer img{
    width: 23px;
    height: 23px;
    filter: invert(1);
}

#fullname{

```

```
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 25px;
    margin: 10px auto;
    margin-top: 60px;
    position: relative;
}

#email{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 25px;
    margin: 20px auto;
    position: relative;
}

#password{
    box-sizing: border-box;
    background-color: aliceblue;
    width: 230px;
    height: 30px;
    padding-top: 5px;
    padding-left: 25px;
    margin: 20px auto;
    position: relative;
}

#button_cont{
    width: 60%;
    position: absolute;
    bottom: 40px;
    left: 20%;
}

#reset_password #button_cont div{
    background-color: rgb(129, 143, 189);
    color: rgb(255, 244, 228);
}
```

```
#button_cont #send_otp_button{
    float: left;
}

#button_cont #reset_password_button{
    float: right;
}

#button_cont div{
    width: 90px;
    height: 30px;
    color: aquamarine;
    background-color: rgb(197, 125, 125);
    margin: 5px;
    padding: 5px;
    box-sizing: border-box;
    text-align: center;
    border-radius: 5px;
    cursor: pointer;
}

#button_cont div:hover{
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);
}

#email_in i{
    position: absolute;
    left: 4px;
    top: 6px;
}

#email i, #password i, #fullname i{
    position: absolute;
    left: 4px;
    top: 6px;
    font-size: 18px;
}

#password i{
    left: 3px;
}
```

Resetpass.js:

```
var file_in;

function swap(self) {
    if(self.id=='reset_password'){
        self.style.opacity="0"
        document.getElementById('send_otp').style.opacity="1"
        setTimeout(()=>{
            document.getElementById('send_otp').style.zIndex=1
            self.style.zIndex=0
        },1000)
    }
    else{
        self.style.opacity="0"
        document.getElementById('reset_password').style.opacity="1"
        setTimeout(()=>{
            document.getElementById('reset_password').style.zIndex=1
            self.style.zIndex=0
        },1000)
    }
}

function sendOtp() {
    var e=$("#email_in input")
    if(e.val().length>0){
        $.ajax({
            type: "POST",
            url: "ServerPage",
            data: {
                type: "FORGOT_PASSWORD SEND_OTP",
                email: e.val()
            },
            success: (json)=>{
                console.log(json)
                if(json.status=="FORGOT_PASSWORD OTP_SENT") {
                    swap(document.getElementById("send_otp"));
                }
            }
        })
    }
}

}
```

```

function reset(){
    var f=$("#fullname input"), e=$("#email input"),p=$("#password input")

    if(e.val().length>0 && p.val().length>0 && f.val().length>0 ){

        $.ajax({
            type: "POST",
            url: "ServerPage",
            data: {  "otp":f.val(),
                    "type":'FORGOT_PASSWORD RESET',
                    "password":e.val()},

            success: (json)=>{
                console.log(json)
                if(json.status=="FORGOT_PASSWORD SUCCESS") {
                    window.location.href='index.jsp';
                }
            },
            error:(e)=>{
                console.log(e);
            }
        })

    }

}

document.addEventListener("DOMContentLoaded",()=>{

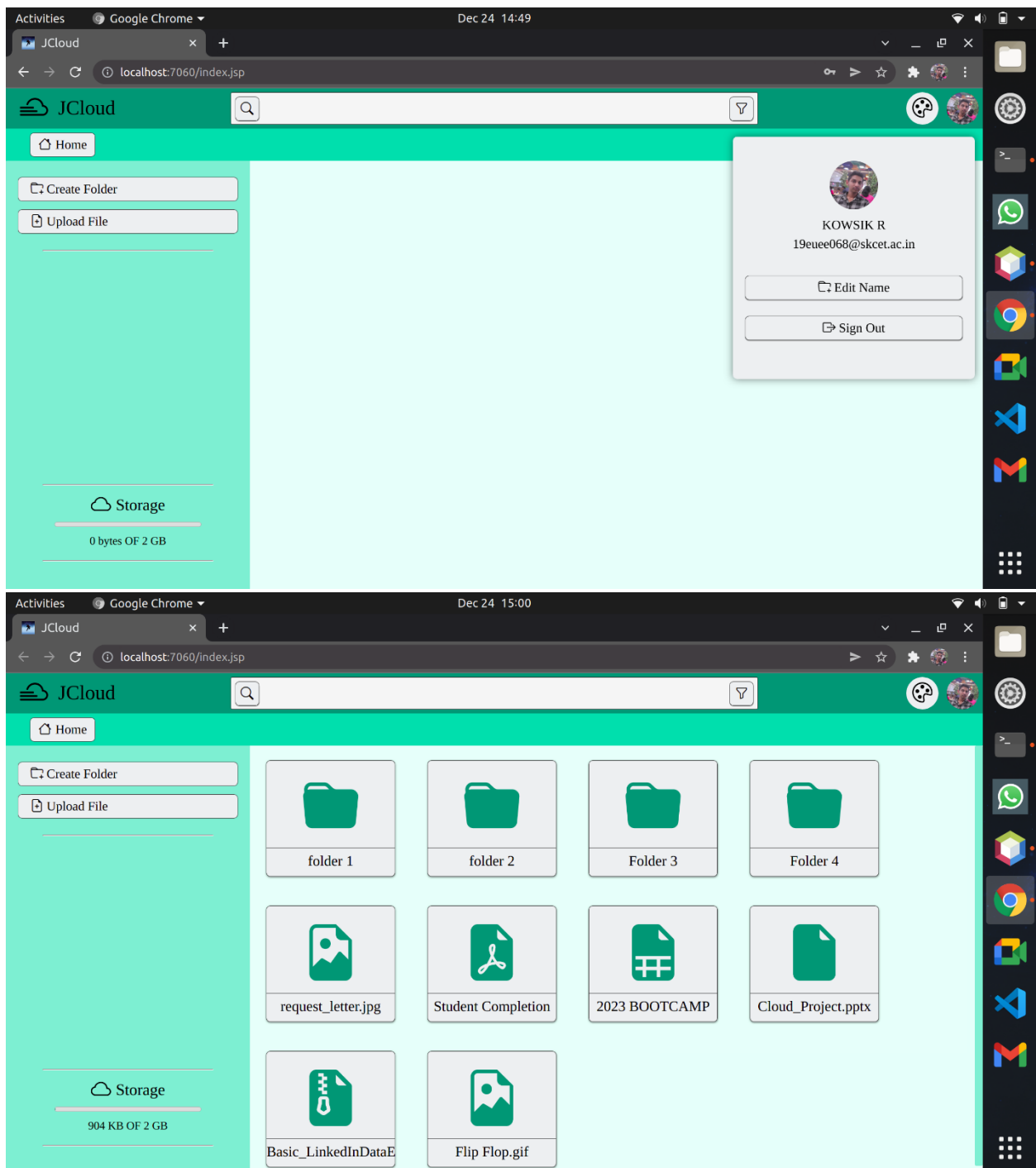
    //code

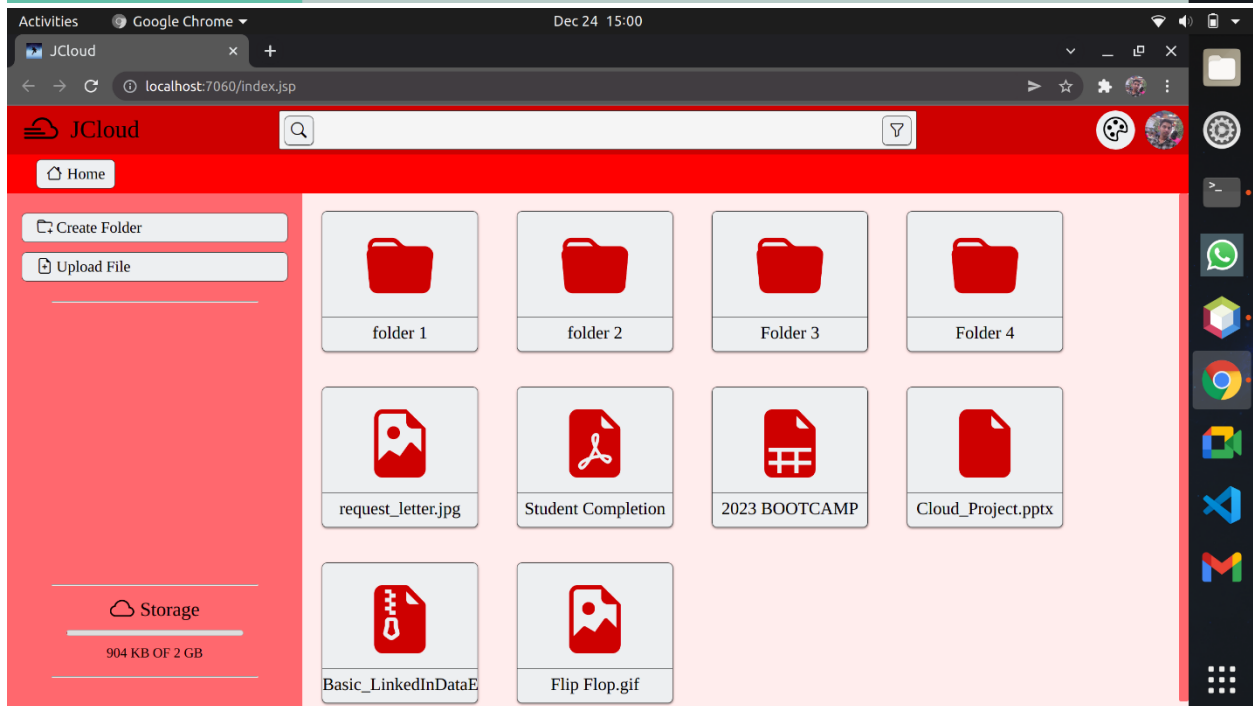
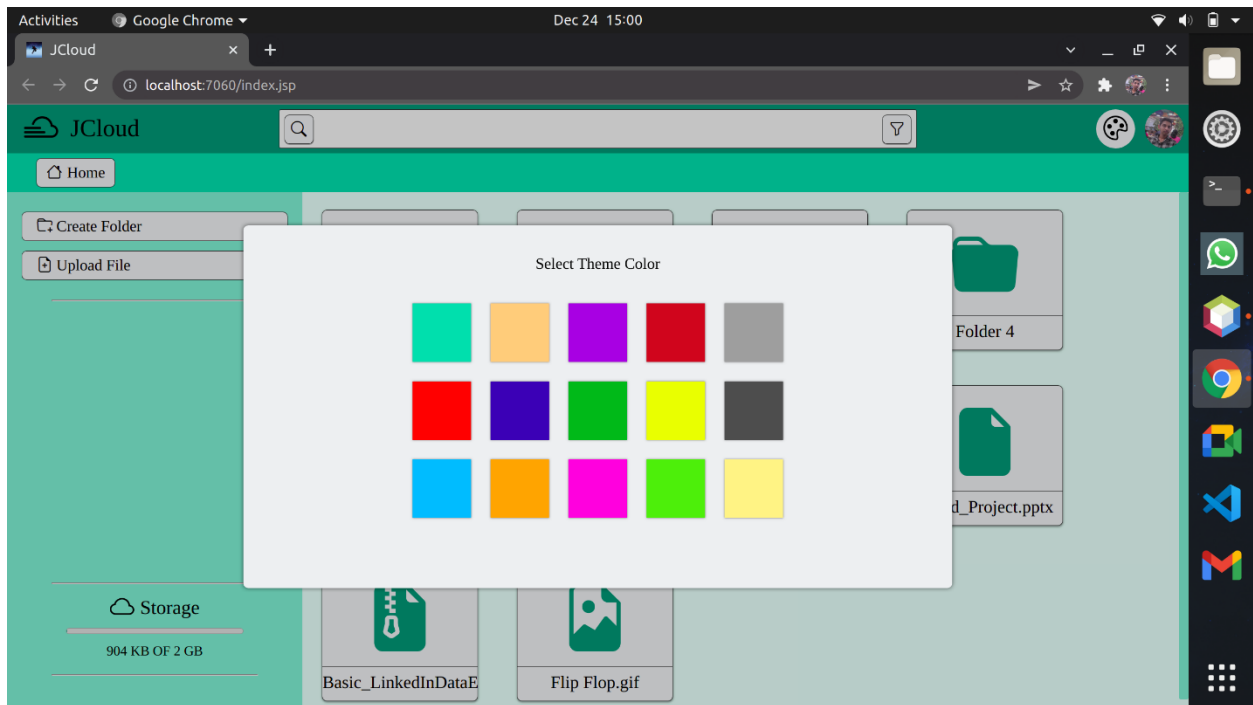
})

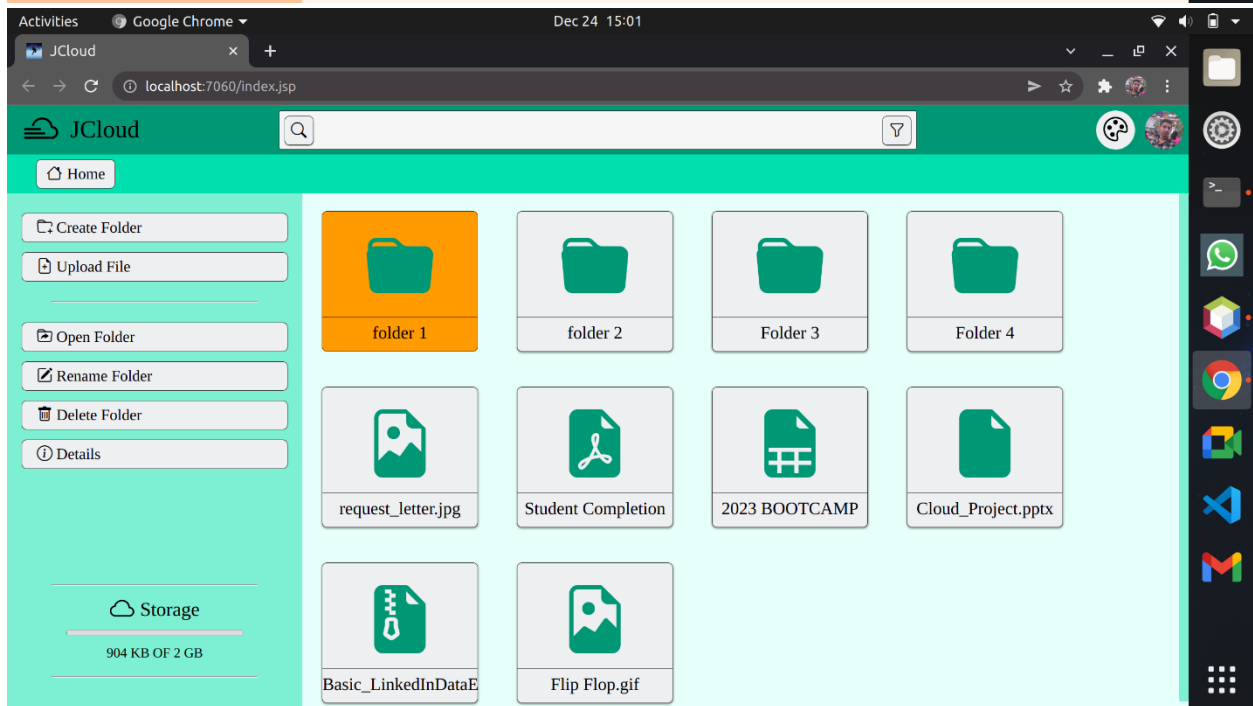
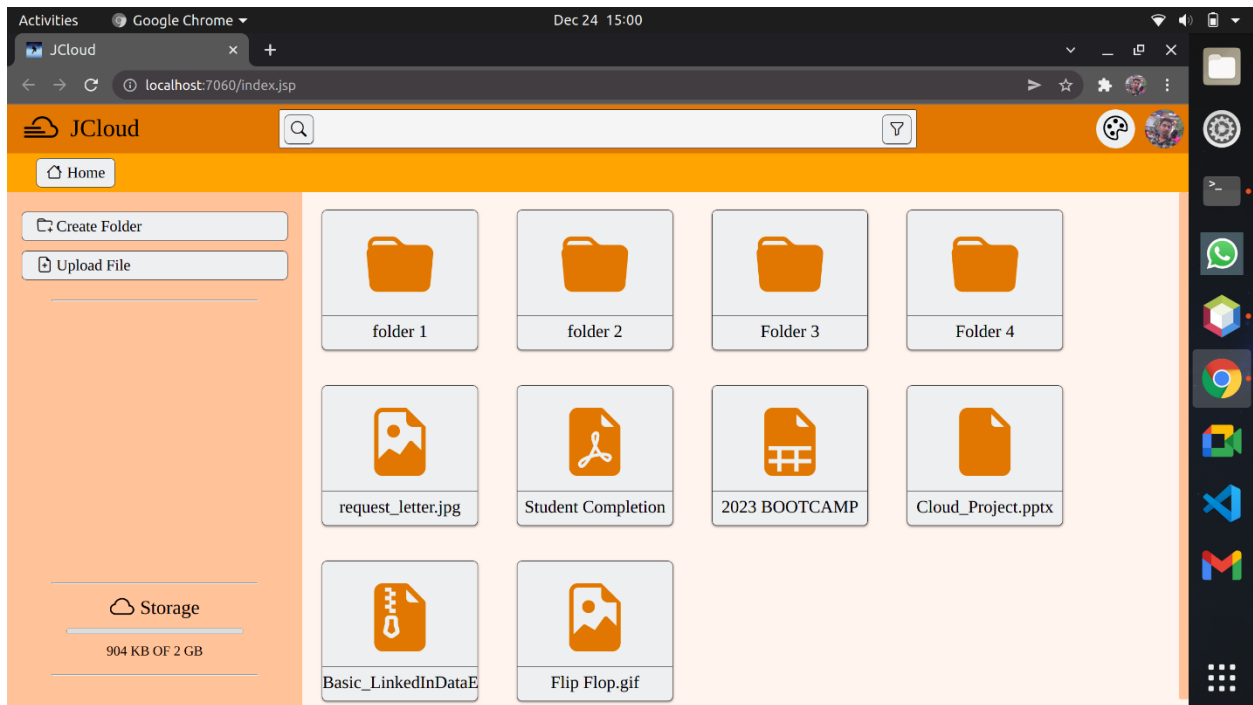
```

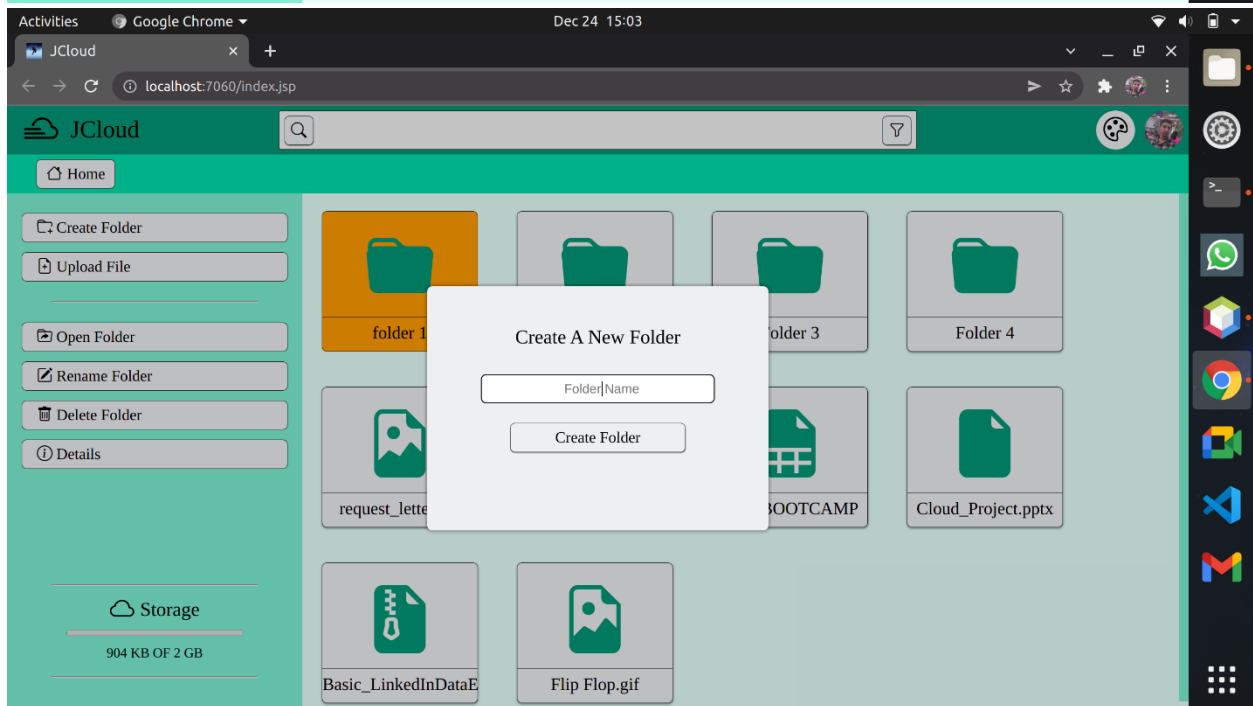
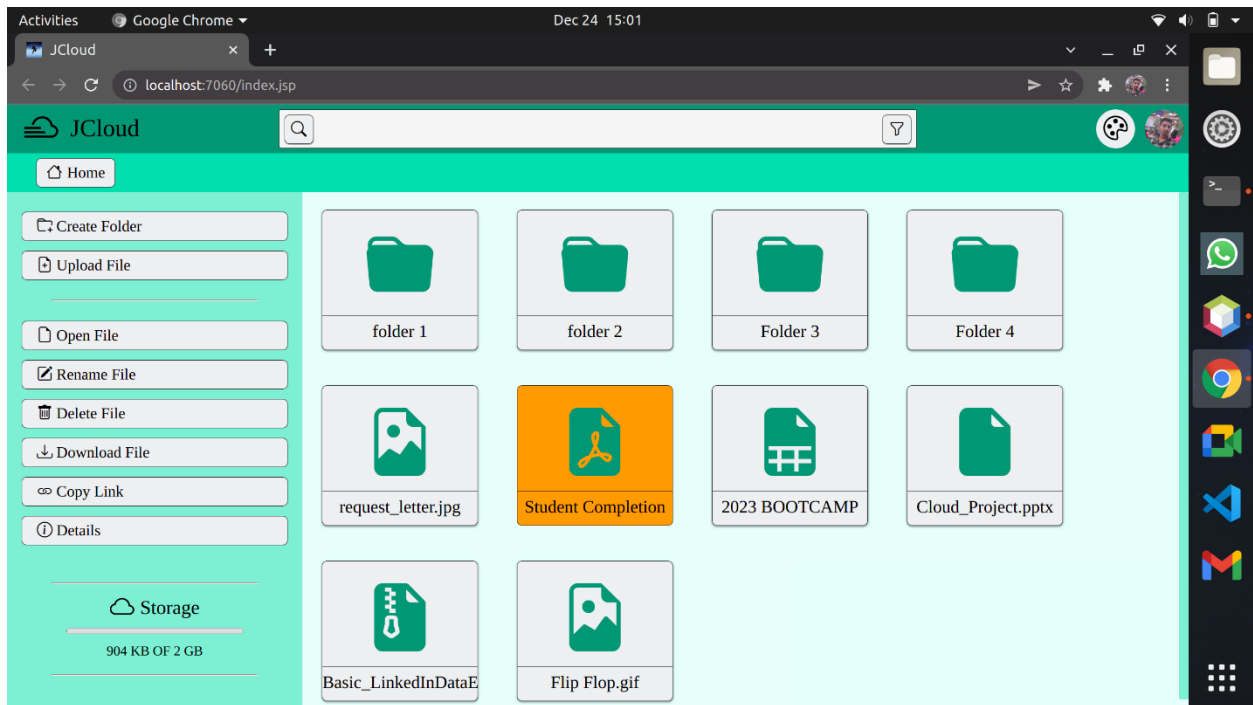

Home Page:

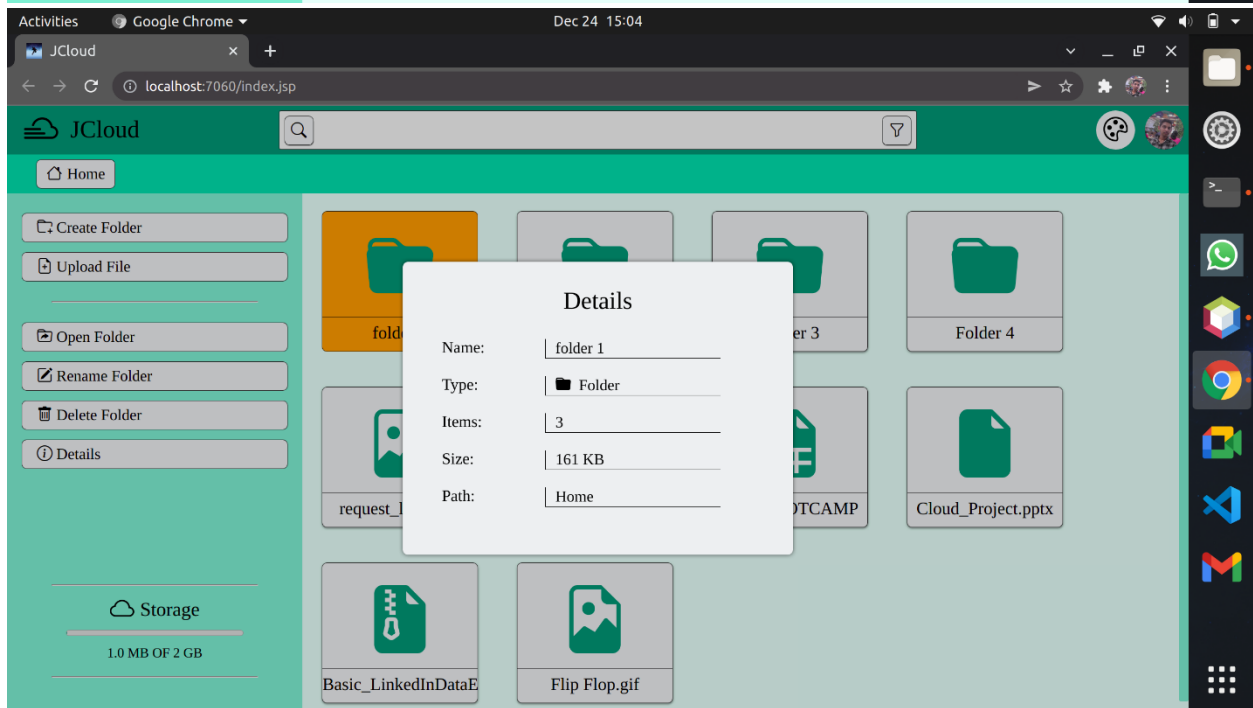
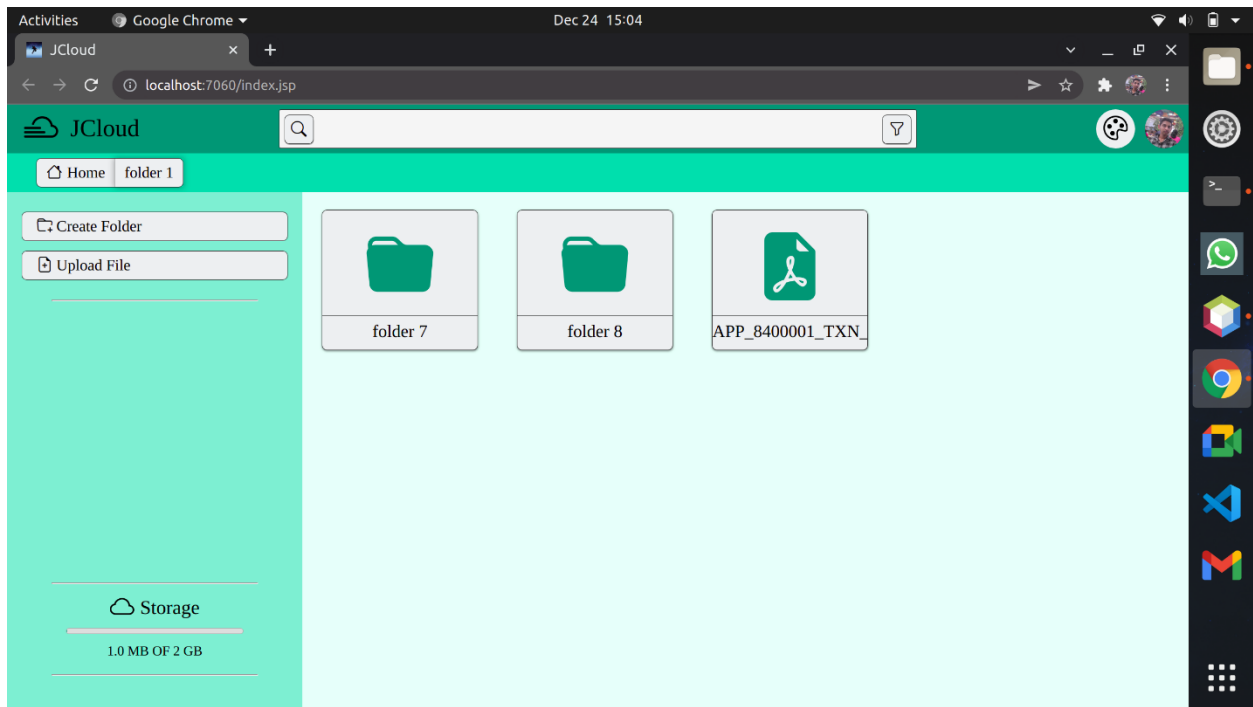
Screenshots:

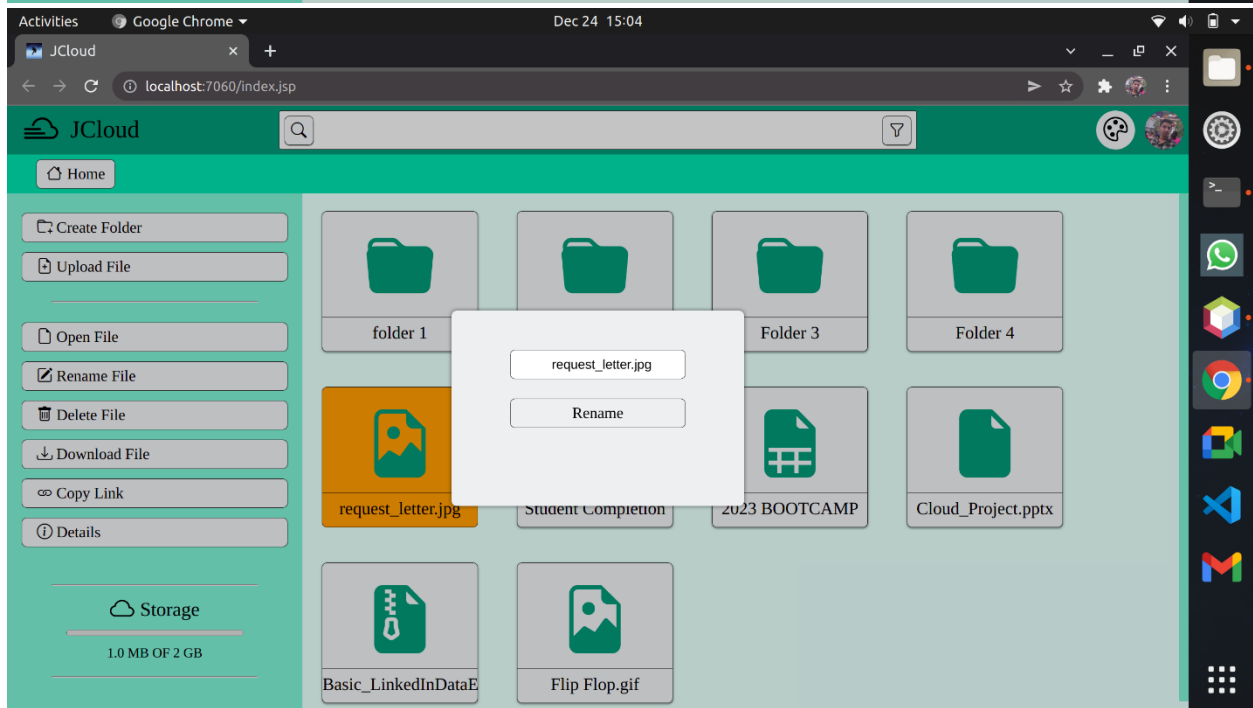
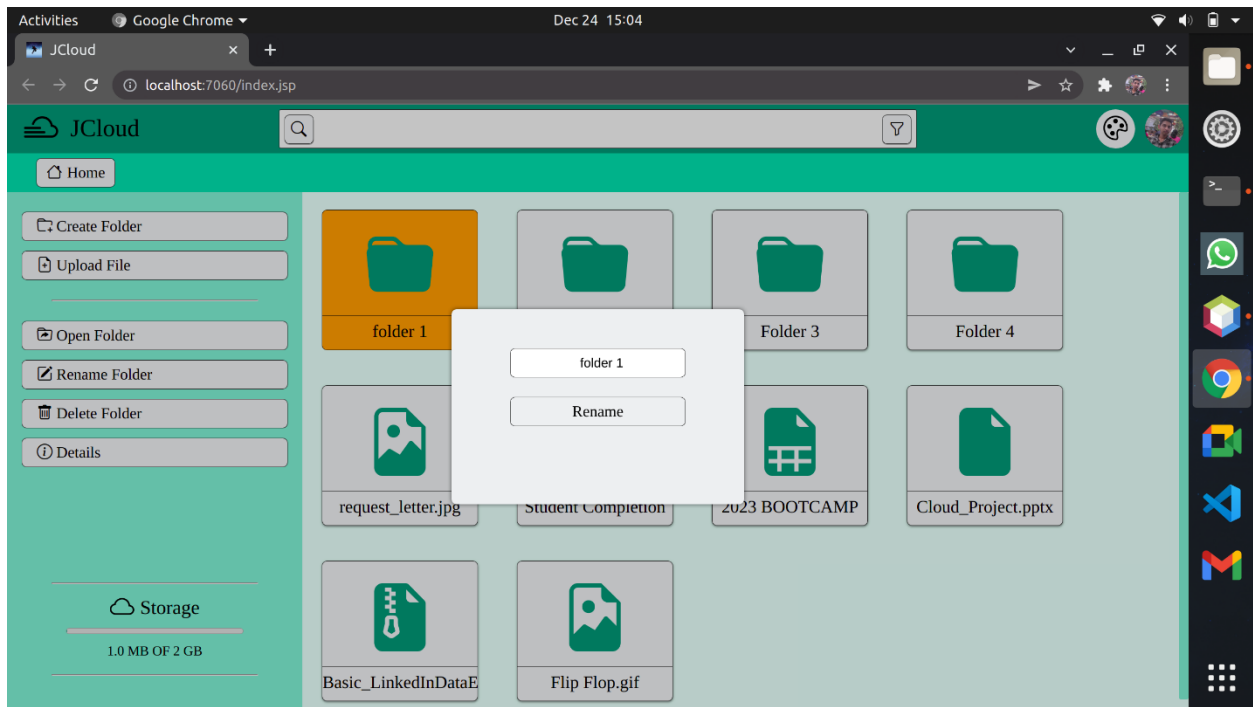


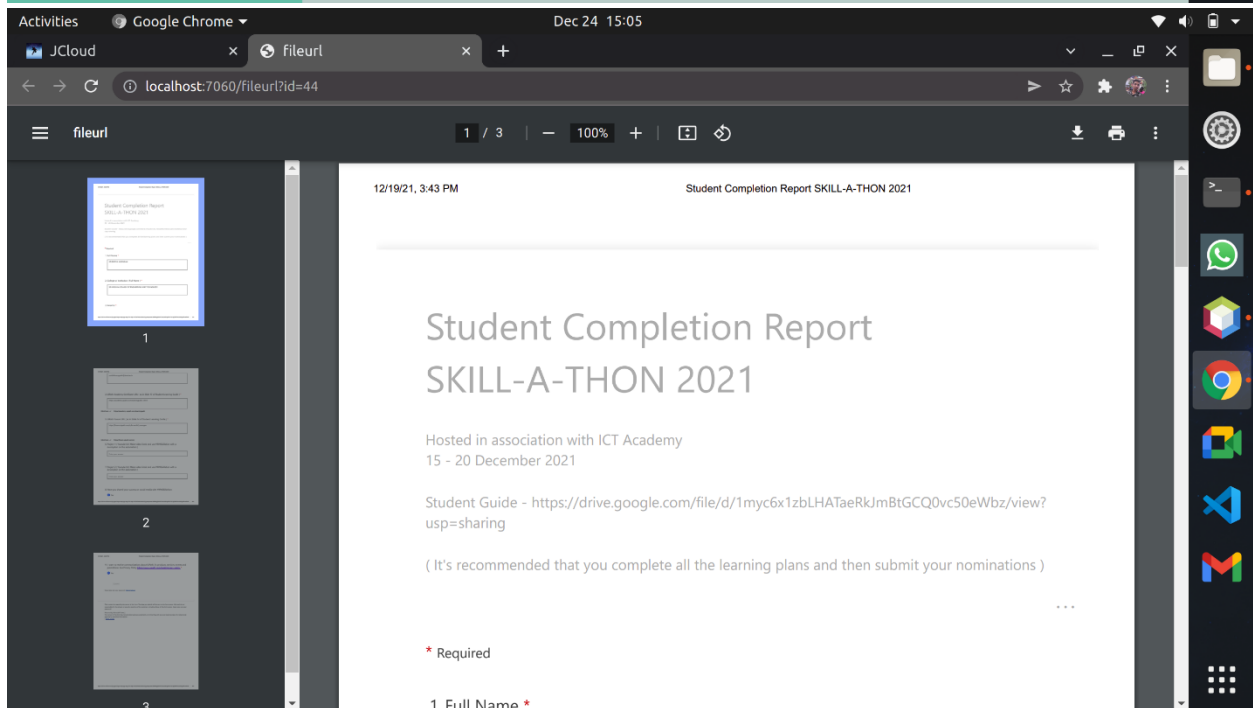
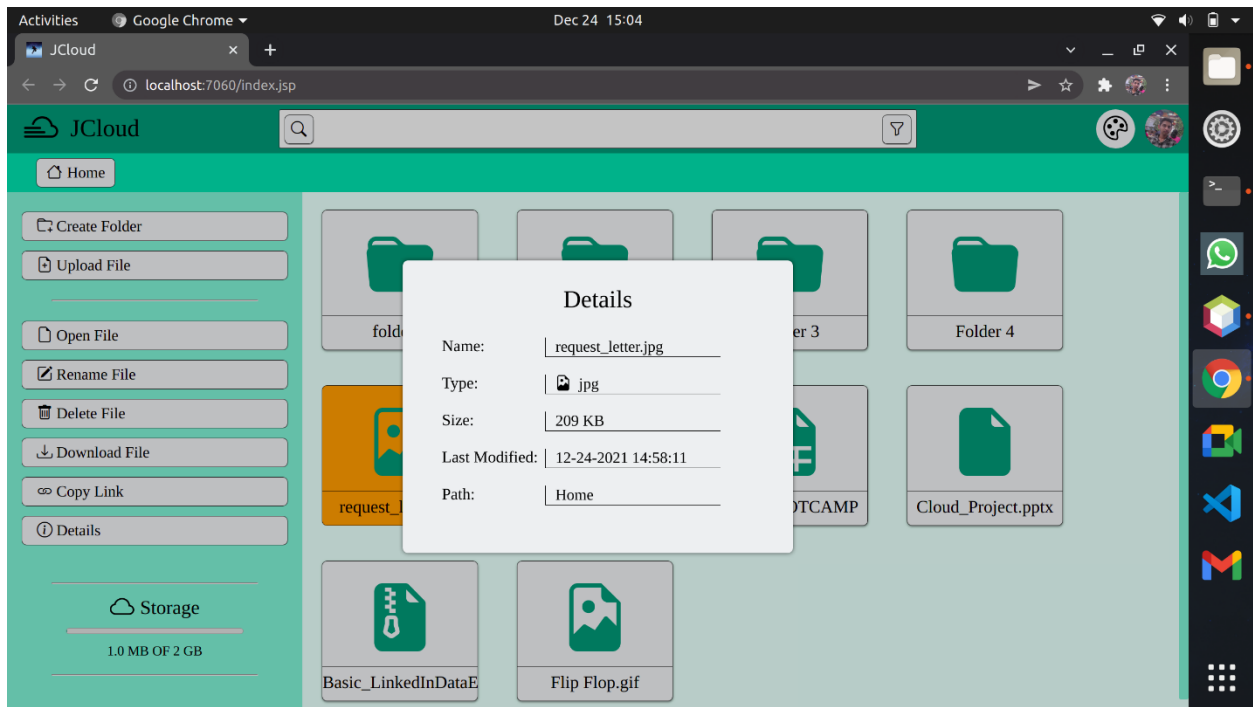


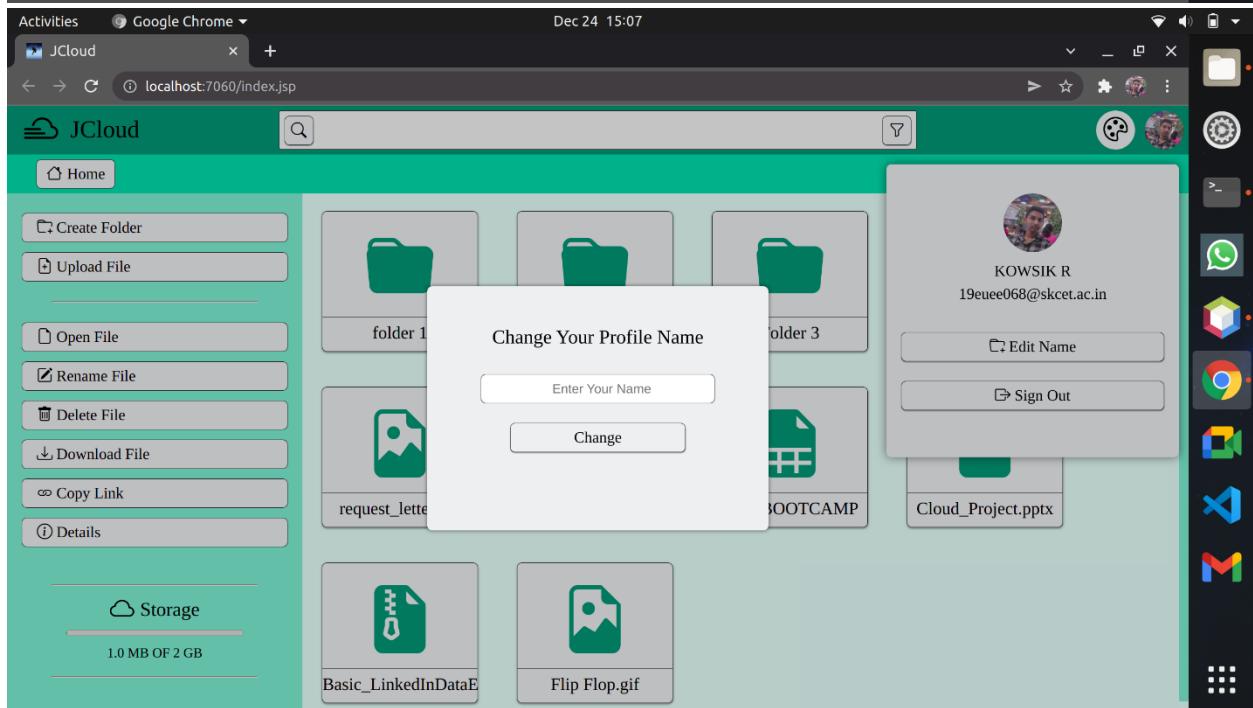
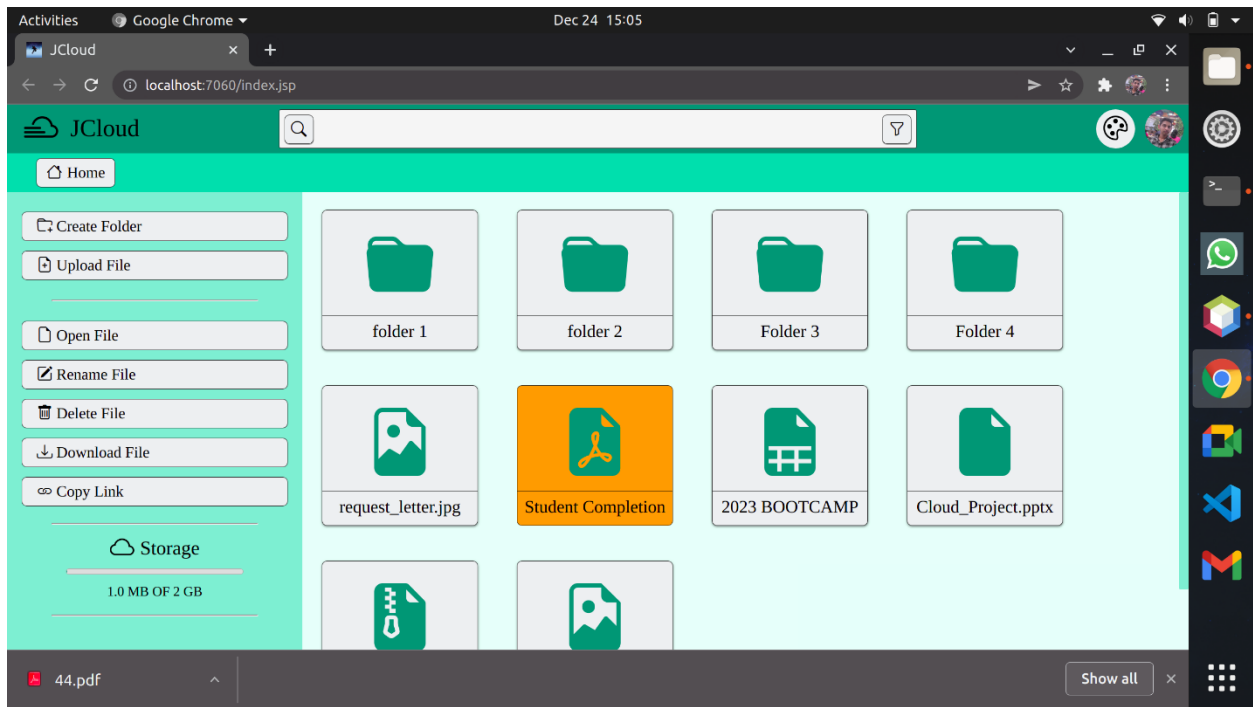


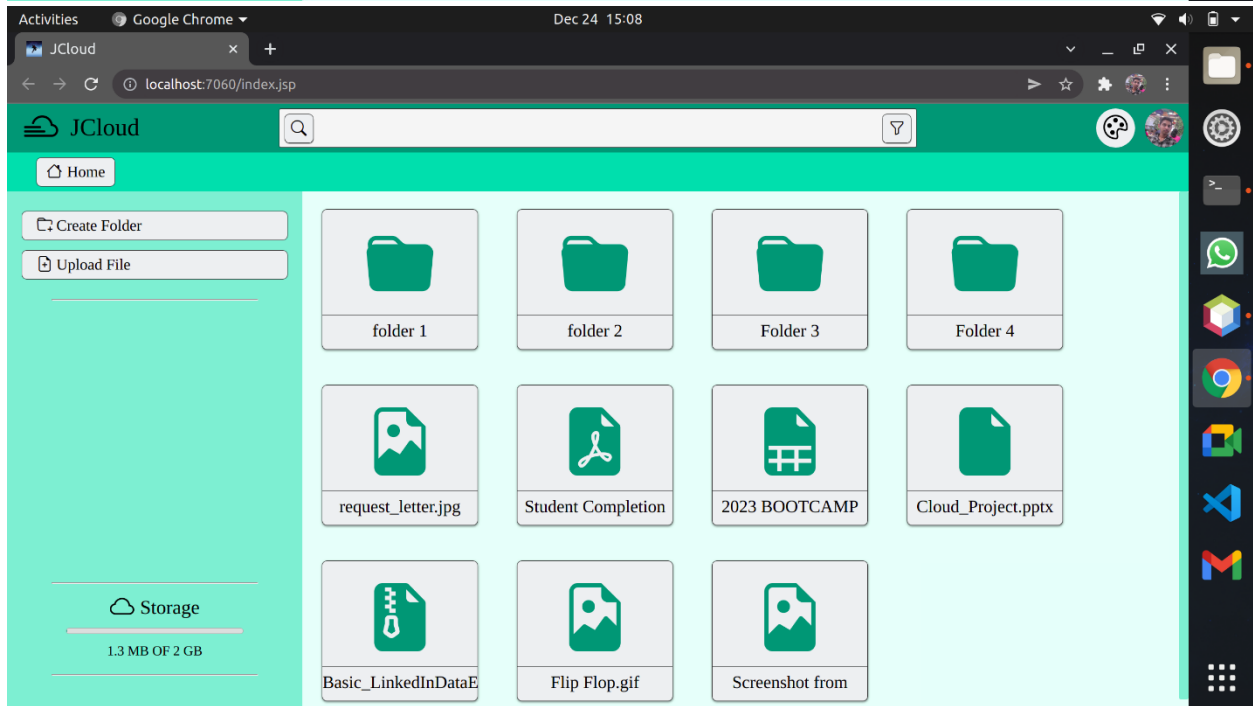
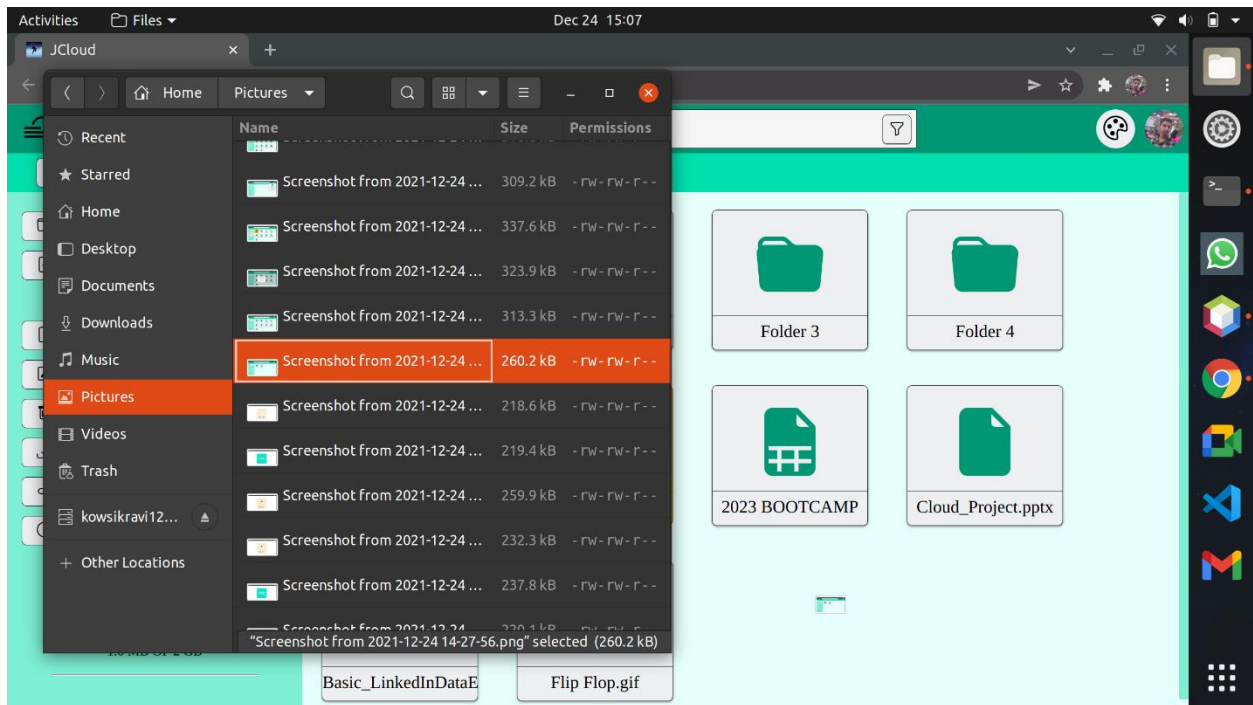












Index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="jcloud.ServerPage" %>
<%@page import="dbms.UserTable" %>
<%@page import="dbms.Dbms" %>
<%
    if(!ServerPage.auth(request)) response.sendRedirect("login.jsp");
    UserTable ut=new UserTable();
%>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JCloud</title>
    <link rel="icon" href="img/spider.jpg"/>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.1/font/bootstrap-icons.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js
"></script>
    <link rel="stylesheet" href="css/style.css">
    <script src="js/script.js"></script>
</head>
<body>
    <div id="nav_bar">
        <div id="left_cont">
            <div id="logo"><i class="bi bi-cloud-fog2"></i></div>
            <div id="logo_name">JCloud</div>
        </div>
        <div id="search_cont">
            <div id="search_box">
                <div id="search_button" aria-label="search"><i class="bi bi-
search"></i></div>
                <input id="search_input" type="text">
                <div id="filter_button" onclick="Shower.show_filter();"><i class=
"bi bi-funnel"></i></div>
                <div id="filter_menu"></div>
            </div>
        </div>
        <div id="right_cont">
            <div id="theme_cont">
```

```

        <div id="theme" onclick="Shower.show_theme();"><i class="bi bi-
palette"></i></div>
    </div>
    <div id="profile_cont">
        <div id="profile" onclick="Shower.show_profile_menu();"><img src=
"ServerPage?type=PROFILE%20PICTURE"></div>
        <div id="profile_menu">
            
            <div id="profile_name"><%= ut.getName(request) %></div>
            <div id="profile_email"><%= Dbms.getCookieValue("email", requ
est) %></div>
            <div id="edit_name" class="menu" onclick="Shower.show_profile
_renamer();">
                <i class="bi bi-folder-plus"></i>
                <span>Edit Name</span>
            </div>
            <div id="reset_password" class="menu" onclick='document.cooki
e="email=";document.cookie="password=";window.location.reload();'>
                <i class="bi bi-box-arrow-right"></i>
                <span>Sign Out</span>
            </div>
        </div>
    </div>
</div>
<div id="path">
    <div id="path_cont">
    </div>
</div>
<div id="container">
    <div id="menu">
        <div id="create_folder" class="menu" onclick="Shower.show_folder_crea
ter();">
            <i class="bi bi-folder-plus"></i>
            <span>Create Folder</span>
        </div>
        <div id="upload_file" class="menu" onclick="CloudFile.uploadFile()">
            <i class="bi bi-file-earmark-plus"></i>
            <span>Upload File</span>
        </div>
    </div>
    <hr>
    <div id="menu_selected">
    </div>

```

```

        <div id="storage_cont">
            <hr>
            <div id="storage_title">
                <i class="bi bi-cloud"></i>
                <div id="storage_word">Storage</div>
            </div>
            <div id="storage"><div id="storage_value" style="width: <%= ut.ge
tStoragePercent(request) %>%;"></div></div>
            <div id="relative_storage" onClick="FolderManager.loadStorage();"
></div>

            <hr>
        </div>
    </div>
    <div id="file_explorer">
    </div>
</div>
<script>
    storage=new FileLoader( <%= ut.getRootId(request) %> );
    $('#relative_storage').text( FileManager.getRelativeSize( <%= ut.getStora
ge(request) %>) +" OF 2 GB" );
</script>
</body>
</html>

```

Style.css:

```

:root{
    --dark: rgb(56, 143, 114);
    --theme: rgb(85, 211, 169);
    --light: rgb(160, 231, 207);
    --lighter: rgb(236, 255, 249);
}

.input{
    width: 180px;
    height: 30px;
    border: 0;
    border-radius: 5px;
    display: block;
    margin: 10px auto;
    border: .3px solid rgba(0, 0, 0, 0.8);
    padding-left: 10px;
    text-align: center;
}

```

```
body{
    margin: 0;
}

*{
    box-sizing: border-box;
}

#relative_storage{
    margin: 10px auto;
    font-size: 14px;
    width: fit-content;
    margin-bottom: 15px;
}

#color_cont{
    display: grid;
    grid-template-columns: auto auto auto auto auto;
    grid-gap: 20px;
    justify-content: center;
}

.color{
    width: 60px;
    height: 60px;
    cursor: pointer;
    box-shadow: 0 0 2px rgba(0, 0, 0, 0.5);
}

.color:hover{
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);
}

#back{
    position: fixed;
    z-index: 10;
    background-color: rgba(0, 0, 0, 0.2);
    width: 100vw;
    height: 100vh;
    top: 0;
    left: 0;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
}

#front{
  border-radius: 5px;
  box-shadow: 0 0 4px rgba(0, 0, 0, 0.6);
  background-color: rgb(237, 239, 241);
  width: 60vw;
  height: 60vh;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 11;
  overflow: hidden;
}

#nav_bar{
  background-color: var(--dark);
  width: 100%;
  height: 50px;
}

#left_cont{
  margin-left: 10px;
  height: 50px;
  width: fit-content;
  color: rgb(0, 0, 0);
}

#logo{
  width: 50px;
  height: 50px;
  display: flex;
  justify-content: center;
  align-items: center;
  float: left;
}

#logo i{
  font-size: 35px;
}

#logo_name{
  float: left;
  font-size: 25px;
}
```

```
padding-top: 9.5px;
margin-left: 5px;
}

#search_cont{
    position: absolute;
    top: 0px;
    left: 20%;
    height: 50px;
    width: 60%;
    display: flex;
    justify-content: center;
    align-items: center;
}

#search_box{
    background-color: whitesmoke;
    width: 90%;
    height: 40px;
    border-radius: 2px;
    position: relative;
    border: .3px solid rgba(0, 0, 0, 0.8);
}

#search_button{
    margin-left: 5px;
    margin-top: 5px;
    width: 30px;
    height: 30px;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color .3s;
    float: left;
    background-color: rgb(237, 239, 241);
    border: .3px solid rgba(0, 0, 0, 0.8);
    box-shadow: .3px 1px 0px rgba(0, 0, 0, 0.3);
}

#search_button:hover{
    box-shadow: .3px 1px 4px rgba(0, 0, 0, 0.3);
}
```

```
#filter_button:hover{
    box-shadow: .3px 1px 4px rgba(0, 0, 0, 0.3);
}

#filter_button{
    margin-right: 5px;
    margin-top: 5px;
    width: 30px;
    height: 30px;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color .3s;
    float: right;
    background-color: rgb(237, 239, 241);
    border: .3px solid rgba(0, 0, 0, 0.8);
    box-shadow: .3px 1px 0px rgba(0, 0, 0, 0.3);
}

#search_input{
    all: unset;
    float: left;
    margin-top: 5px;
    margin-left: 5px;
    height: 30px;
    width: calc(100% - 80px);
}

#right_cont{
    float: right;
    position: absolute;
    top: 0;
    right: 0;
}

#profile_cont{
    width: 50px;
    height: 50px;
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}
```



```
#profile:hover{
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}

#profile{
    background-color: rgb(237, 239, 241);
    width: 40px;
    height: 40px;
    border-radius: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}

#profile_menu{
    width: 300px;
    height: 300px;
    background-color: rgb(237, 239, 241);
    position: fixed;
    top: 60px;
    right: 10px;
    display: none;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
    z-index: 5;
    border-radius: 5px;
}

#profile_img{
    width: 100%;
    height: 100%;
    overflow: hidden;
    border-radius: 50%;
}

#theme_cont{
    width: 50px;
    height: 50px;
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
#theme{
  width: 40px;
  height: 40px;
  border-radius: 50%;
  background-color: rgb(237, 239, 241);
  cursor: pointer;
  font-size: 25px;
  display: flex;
  justify-content: center;
  align-items: center;
}

#theme:hover{
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}

#path{
  width: 100%;
  height: 40px;
  background-color: var(--theme);
  padding-top: 5px;
}

#path_cont{
  width: fit-content;
  margin-left: 30px;
  border-radius: 5px;
  height: 30px;
  background-color: rgb(237, 239, 241);
  border: .3px solid rgba(0, 0, 0, 0.8);
}

#home{
  width: 80px;
}

#home i{
  position: relative;
  top: 5px;
  left: 10px;
}

#home span{
  position: relative;
  top: 5px;
```

```
    left: 15px;
    padding: 0;
}

.folder_path span{
    position: relative;
    top: 5px;
    padding: 0 10px;
}

.folder_path:hover{
    box-shadow: 0px 0px 6px rgba(0, 0, 0, 0.4);
}

.folder_path{
    float: left;
    border-right: .3px solid rgba(0, 0, 0, 0.8);
    height: 100%;
    cursor: pointer;
}

.folder_path:last-child{
    border-right: 0;
}

.menu{
    background-color: rgb(237, 239, 241);
    width: 90%;
    height: 30px;
    margin: 10px auto;
    border-radius: 5px;
    border: .3px solid rgba(0, 0, 0, 0.8);
    padding: 5px 15px;
    cursor: pointer;
    box-shadow: 0px 0px 0px rgba(0, 0, 0, 0.4);
}

#menu hr{
    width: 70%;
    margin: 20px auto;
}

#storage_cont hr{
    margin: 0 auto;
    margin-bottom: 10px;
}
```

```
}

.menu:hover{
    box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.4);
}

#container{
    background-color: rgb(235, 225, 240);
    width: 100%;
    height: calc(100vh - 90px);
}

#menu{
    float: left;
    width: 25%;
    height: 100%;
    background-color: var(--light);
    position: relative;
    padding-top: 10px;
    overflow: hidden;
}

#file_explorer{
    width: 75%;
    height: 100%;
    background-color: var(--lighter);
    float: left;
    overflow-y: scroll;
    overflow-x: hidden;
}

#file_explorer::-webkit-scrollbar{
    background-color: transparent;
    width: 10px;
}

#file_explorer::-webkit-scrollbar-thumb{
    background-color: var(--light);
    border-radius: 2px;
}

#storage_cont{
    width: 100%;
    height: 140px;
    position: absolute;
```

```
    bottom: 0px;
    padding-top: 10px;
    color: rgb(15, 15, 15);
    background-color: var(--light);
}

#storage_title{
    height: fit-content;
    width: fit-content;
    margin: 0 auto;
}

#storage_title i{
    font-size: 25px;
    float: left;
}

#storage_word{
    font-size: 20px;
    float: left;
    margin-left: 6px;
    margin-top: 2px;
}

#storage{
    width: 60%;
    height: 6px;
    background-color: rgb(221, 221, 221);
    margin: 45px auto;
    margin-bottom: 10px;
    border-radius: 10px;
    overflow: hidden;
    border: .1px solid rgba(0, 0, 0, 0.2);
}

#storage_value{
    width: 10%;
    height: 100%;
    background-color: var(--dark);
    border-radius: 6px;
}

#new_cont{
    width: 100%;
    height: 80px;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
}

#filter_menu{
    position: absolute;
    top: 40px;
    z-index: 1;
    width: 100%;
    height: 300px;
    background-color: inherit;
    border: .3px solid rgba(0, 0, 0, 0.8);
    border-bottom-left-radius: 10px;
    border-bottom-right-radius: 10px;
    box-shadow: .5px 2px 5px rgba(0, 0, 0, 0.5);
    display: none;
}

.f_cont{
    width: 200px;
    height: 180px;
    display: flex;
    justify-content: center;
    align-items: center;
    float: left;
}

.f_inner_cont{
    width: 80%;
    height: 80%;
    background-color: rgb(237, 239, 241);
    padding-top: 10px;
    border-radius: 5px;
    border: .3px solid rgba(0, 0, 0, 0.8);
    position: relative;
    box-shadow: .2px 1px 2px rgba(0, 0, 0, 0.5);
    cursor: pointer;
}

.f_inner_cont:hover{
    box-shadow: 1px 2px 5px rgba(0, 0, 0, 0.6);
}

.f_icon{
```

```
    margin-top: 10px;
    font-size: 70px;
    text-align: center;
    color: var(--dark);
}

.f_name{
    font-size: large;
    text-align: center;
    position: absolute;
    bottom: 5px;
    width: 100%;
    height: 30px;
    border-top: .3px solid rgba(0, 0, 0, 0.8);
    padding-top: 5px;
    overflow: hidden;
}

.f_selected .f_inner_cont{
    background-color: rgb(255, 153, 0);
    box-shadow: none;
}

#profile_img{
    width: 60px;
    height: 60px;
    border-radius: 50%;
    background-color: var(--dark);
    margin: 10px auto;
    margin-top: 30px;
    display: block;
    overflow: hidden;
}

#profile_name{
    margin: 0 auto;
    text-align: center;
}

#profile_email{
    margin: 5px auto;
    text-align: center;
}

#edit_name{
```

```
    text-align: center;
    margin-top: 30px;
    margin-bottom: 20px;
    box-shadow: 0.2px 1px 1px rgba(0, 0, 0, 0.4);
}

#reset_password{
    text-align: center;
    box-shadow: 0.2px 1px 1px rgba(0, 0, 0, 0.4);
}

#edit_name:hover{
    box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.4);
}

#reset_password:hover{
    box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.4);
}

.box_shadow{
    box-shadow: 0.2px 1px 1px rgba(0, 0, 0, 0.4);
}

.property_cont{
    width: 90%;
    height: 20px;
    margin: 18px auto;
}

.property_cont .key{
    width: 35%;
    height: 100%;
    float: left;
    padding-left: 20px;
    font-size: 16.5px;
}

.property_cont .value i{
    margin-right: 8px;
}

.property_cont .value{
    width: 50%;
    height: 100%;
    float: left;
    border-bottom: .5px solid black;
    border-left: .5px solid black;
    padding: 0px 10px;
```



```
    overflow: hidden;
}
```

Script.js:

```
var storage;

class Shower{
    static is_shown=false;
    static is_filter_shown=false;

    static set_back(){
        var ele=document.createElement("div")
        ele.id="back"
        ele.addEventListener("click",()=>{
            ele.remove()
            document.getElementById("front").remove()
        })
        document.body.append(ele)
    }

    static remove_back(){
        document.getElementById("back").remove()
        document.getElementById("front").remove()
    }

    static set_theme(v){
        var root=document.querySelector(':root').style
        switch (v) {
            case 2:
                root.setProperty('--dark','rgb(194, 151, 100)')
                root.setProperty('--theme','rgb(255, 199, 130)')
                root.setProperty('--light','rgb(255, 228, 196)')
                root.setProperty('--lighter','rgb(255, 246, 235)')
                break;
            case 3:
                root.setProperty('--dark','rgb(88, 22, 150)')
                root.setProperty('--theme','rgb(138, 43, 226)')
                root.setProperty('--light','rgb(213, 168, 255)')
                root.setProperty('--lighter','rgb(248, 242, 255)')
                break;
            case 4:
                root.setProperty('--dark','rgb(134, 22, 22)')
                root.setProperty('--theme','rgb(187, 52, 52)')
```

```
        root.setProperty('--light', 'rgb(214, 103, 103)')
        root.setProperty('--lighter', 'rgb(248, 237, 237)')
        break;
    case 5:
        root.setProperty('--dark', 'rgb(100, 100, 100)')
        root.setProperty('--theme', 'rgb(158, 158, 158)')
        root.setProperty('--light', 'rgb(201, 201, 201)')
        root.setProperty('--lighter', 'rgb(240, 240, 240)')
        break;
    case 6:
        root.setProperty('--dark', 'rgb(184, 0, 0)')
        root.setProperty('--theme', 'rgb(255, 0, 0)')
        root.setProperty('--light', 'rgb(255, 122, 122)')
        root.setProperty('--lighter', 'rgb(255, 238, 238)')
        break;
    case 7:
        root.setProperty('--dark', 'rgb(0, 0, 129)')
        root.setProperty('--theme', 'rgb(0, 0, 180)')
        root.setProperty('--light', 'rgb(0, 0, 255)')
        root.setProperty('--lighter', 'rgb(235, 235, 255)')
        break;
    case 8:
        root.setProperty('--dark', 'rgb(0, 128, 0)')
        root.setProperty('--theme', 'rgb(0, 167, 0)')
        root.setProperty('--light', 'rgb(122, 240, 122)')
        root.setProperty('--lighter', 'rgb(232, 255, 232)')
        break;
    case 9:
        root.setProperty('--dark', 'rgb(207, 207, 0)')
        root.setProperty('--theme', 'rgb(243, 243, 0)')
        root.setProperty('--light', 'rgb(255, 255, 160)')
        root.setProperty('--lighter', 'rgb(255, 255, 227)')
        break;
    case 10:
        root.setProperty('--dark', 'rgb(51, 51, 51)')
        root.setProperty('--theme', 'rgb(77, 77, 77)')
        root.setProperty('--light', 'rgb(116, 116, 116)')
        root.setProperty('--lighter', 'rgb(233, 233, 233)')
        break;
    case 11:
        root.setProperty('--dark', 'rgb(0, 127, 212)')
        root.setProperty('--theme', 'rgb(0, 195, 255)')
        root.setProperty('--light', 'rgb(143, 229, 255)')
        root.setProperty('--lighter', 'rgb(235, 250, 255)')
        break;
```

```

        case 12:
            root.setProperty('--dark', 'rgb(209, 118, 0)')
            root.setProperty('--theme', 'rgb(255, 161, 37)')
            root.setProperty('--light', 'rgb(255, 194, 159)')
            root.setProperty('--lighter', 'rgb(255, 245, 239)')
            break;
        case 13:
            root.setProperty('--dark', 'rgb(168, 0, 173)')
            root.setProperty('--theme', 'rgb(224, 1, 224)')
            root.setProperty('--light', 'rgb(255, 173, 251)')
            root.setProperty('--lighter', 'rgb(254, 239, 255)')
            break;
        case 14:
            root.setProperty('--dark', 'rgb(118, 182, 0)')
            root.setProperty('--theme', 'rgb(141, 218, 0)')
            root.setProperty('--light', 'rgb(182, 255, 47)')
            root.setProperty('--lighter', 'rgb(247, 255, 231)')
            break;
        case 15:
            root.setProperty('--dark', 'rgb(255, 225, 104)')
            root.setProperty('--theme', 'rgb(255, 232, 138)')
            root.setProperty('--light', 'rgb(255, 243, 195)')
            root.setProperty('--lighter', 'rgb(255, 253, 246)')
            break;
        default:
            root.setProperty('--dark', 'rgb(56, 143, 114)')
            root.setProperty('--theme', 'rgb(85, 211, 169)')
            root.setProperty('--light', 'rgb(160, 231, 207)')
            root.setProperty('--lighter', 'rgb(236, 255, 249)')
            break;
    }
}

static show_theme(){
    this.set_back()
    var ele=document.createElement("div")
    ele.id="front"
    ele.innerHTML=`<div style="margin: 10px auto;width: fit-
content;height: 40px;margin-top: 30px;">Select Theme Color</div>
    <div style="margin: 10px auto;width: 75%;height: fit-
content;" id="color_cont">
        <div onclick="Shower.set_theme(1);" class="color" style="background-
color: rgb(85, 211, 169);"></div>
        <div onclick="Shower.set_theme(2);" class="color" style="background-
color: rgb(255, 199, 130);"></div>

```

```

        <div onclick="Shower.set_theme(3);" class="color" style="background-
color: rgb(138, 43, 226);"></div>
        <div onclick="Shower.set_theme(4);" class="color" style="background-
color: rgb(187, 52, 52);"></div>
        <div onclick="Shower.set_theme(5);" class="color" style="background-
color: rgb(158, 158, 158);"></div>
        <div onclick="Shower.set_theme(6);" class="color" style="background-
color: rgb(255, 0, 0);"></div>
        <div onclick="Shower.set_theme(7);" class="color" style="background-
color: rgb(0, 0, 180);"></div>
        <div onclick="Shower.set_theme(8);" class="color" style="background-
color: rgb(0, 167, 0);"></div>
        <div onclick="Shower.set_theme(9);" class="color" style="background-
color: rgb(243, 243, 0);"></div>
        <div onclick="Shower.set_theme(10);" class="color" style="background-
color: rgb(77, 77, 77);"></div>
        <div onclick="Shower.set_theme(11);" class="color" style="background-
color: rgb(0, 195, 255);"></div>
        <div onclick="Shower.set_theme(12);" class="color" style="background-
color: rgb(255, 161, 37);"></div>
        <div onclick="Shower.set_theme(13);" class="color" style="background-
color: rgb(224, 1, 224);"></div>
        <div onclick="Shower.set_theme(14);" class="color" style="background-
color: rgb(141, 218, 0);"></div>
        <div onclick="Shower.set_theme(15);" class="color" style="background-
color: rgb(255, 232, 138);"></div>
    </div>`
    document.body.append(ele)
}
static show_profile_menu(){
    if(Shower.is_shown){
        document.getElementById("profile_menu").style.display="none";
        Shower.is_shown=false;
    }
    else {
        document.getElementById("profile_menu").style.display="block";
        Shower.is_shown=true;
    }
}
static show_filter(){
    if (Shower.is_filter_shown) {
        document.getElementById("filter_menu").style.display="none"
        Shower.is_filter_shown=false;
    }
    else{

```

```

        document.getElementById("filter_menu").style.display="block"
        Shower.is_filter_shown=true;
    }
}
static show_folder_renamer(id,name){
    this.set_back()
    var ele=document.createElement("div")
    ele.id="front"
    ele.style.width='300px'
    ele.style.height='200px'
    ele.innerHTML=`<input type="text" placeholder="Enter New Folder Name" class="input" style="margin-top: 40px;" value="\`+name+\`">
    <div class="menu" style="width: 180px;text-align: center; margin-top: 20px;" onclick="FolderManager.renameFolder($('#front input').val(),\`+id+\`);">Rename</div>`
    document.body.append(ele)
}
static show_folder_creator(){
    this.set_back()
    var ele=document.createElement("div")
    ele.id="front"
    ele.style.width='350px'
    ele.style.height='250px'
    ele.innerHTML=
    `<div style="width: fit-content; margin: 20px auto; height: 30px; font-size: 20px; margin-top: 40px;">Create A New Folder</div>
    <input type="text" pattern="."+ class="input" style="width: 240px;" placeholder="Folder Name">
    <div class="menu box_shadow" style="width: 180px; text-align: center; margin-top: 20px;" onclick="FolderManager.createFolder()">Create Folder</div>`
    document.body.append(ele)
}
static show_profile_renamer(){
    this.set_back()
    var ele=document.createElement("div")
    ele.id="front"
    ele.style.width='350px'
    ele.style.height='250px'
    ele.innerHTML=
    `<div style="width: fit-content; margin: 20px auto; height: 30px; font-size: 20px; margin-top: 40px;">Change Your Profile Name</div>
    <input type="text" pattern="."+ class="input" style="width: 240px;" placeholder="Enter Your Name">`
}

```

```

        <div class="menu box_shadow" style="width: 180px; text-align: center; margin-top: 20px; " onclick="ProfileManager.changeName();">Change</div>`
        document.body.append(ele)
    }
    static show_file_renamer(id,name){
        this.set_back()
        var ele=document.createElement("div")
        ele.id="front"
        ele.style.width='300px'
        ele.style.height='200px'
        ele.innerHTML=`<input type="text" placeholder="Enter New File Name" class="input" style="margin-top: 40px;" value="+name+">
        <div class="menu" style="width: 180px;text-align: center; margin-top: 20px;" onclick="FileManager.renameFile($('#front input').val(),`+id+`);">Rename</div>`
        document.body.append(ele)
    }
    static show_folder_details(name,id){
        this.set_back()
        var f=storage.getFolderById(id,storage.currentFolder)
        var ele=document.createElement("div")
        ele.id="front"
        ele.style.width='400px'
        ele.style.height='300px'
        ele.innerHTML=
        `<div style="margin: 25px auto; width: fit-content; font-size: 25px;">Details</div>
        <div class="property_cont">
            <div class="key">Name:</div>
            <div class="value">`+name+`</div>
        </div>
        <div class="property_cont">
            <div class="key">Type:</div>
            <div class="value"><i class="bi bi-folder-fill"></i>Folder</div>
        </div>
        <div class="property_cont">
            <div class="key">Items:</div>
            <div class="value"></div>
        </div>
        <div class="property_cont">
            <div class="key">Size:</div>
            <div class="value"></div>
        </div>
        <div class="property_cont">

```

```

        <div class="key">Path:</div>
        <div class="value">`+PathLoader.getPath()+`</div>
    </div>`
    document.body.append(ele)
    if(storage.loadedIds.includes(id)) {
        $(' .value')[2].innerText=(f.files.length+f.folders.length)
    }
    else{
        storage.setFolder(f,()=>{
            $(' .value')[2].innerText=(f.files.length+f.folders.length)
        })
    }
    FolderManager.loadFolderSize(id)

}
static show_file_details(id,name){
    this.set_back()
    var ele=document.createElement("div")
    ele.id="front"
    ele.style.width='400px'
    ele.style.height='300px'
    ele.innerHTML=
    `<div style="margin: 25px auto; width: fit-content; font-
size: 25px;">Details</div>
    <div class="property_cont">
        <div class="key">Name:</div>
        <div class="value">`+name+`</div>
    </div>
    <div class="property_cont">
        <div class="key">Type:</div>
        <div class="value"><i class="bi `+CloudFile.ext_map[CloudFile.getExt(
name)]+` "></i>`+CloudFile.getExt(name)+`</div>
    </div>
    <div class="property_cont">
        <div class="key">Size:</div>
        <div class="value"></div>
    </div>
    <div class="property_cont">
        <div class="key">Last Modified:</div>
        <div class="value"></div>
    </div>
    <div class="property_cont">
        <div class="key">Path:</div>
        <div class="value">`+PathLoader.getPath()+`</div>
    </div>`

```

```

        document.body.append(ele)
        FileManager.loadInfos(id,name);
    }
}

class ProfileManager{
    static changeName(){
        document.body.style.cursor='wait'
        $.ajax({
            type: "POST",
            url: "ServerPage",
            data: { "type" : "UPDATE FULLNAME", "name": $('#front input').val() }
        },

        success: (json)=>{
            document.body.style.cursor='default'
            $('#profile_name').text($('#front input').val())
        },
        error: (e)=>{
            document.body.style.cursor='default'
        }
    })
}

class CloudFile{
    static ext_map={
        "jpg": "bi-file-earmark-image",
        "jpeg": "bi-file-earmark-image",
        "png": "bi-file-earmark-image",
        "gif": "bi-file-earmark-image",
        "svg": "bi-file-earmark-image",
        "mp4": "bi-file-earmark-play-fill",
        "mpeg4": "bi-file-earmark-play-fill",
        "avi": "bi-file-earmark-play-fill",
        "mov": "bi-file-earmark-play-fill",
        "mpeg2": "bi-file-earmark-play-fill",
        "pdf": "bi-file-earmark-pdf-fill",
        "doc": "bi-file-earmark-word-fill",
        "docx": "bi-file-earmark-word-fill",
        "txt": "bi-file-earmark-text-fill",
        "html": "bi-file-earmark-code-fill",
        "css": "bi-file-earmark-code-fill",
        "js": "bi-file-earmark-code-fill",
    }
}

```



```

    "py": "bi-file-earmark-code-fill",
    "java": "bi-file-earmark-code-fill",
    "pyc": "bi-file-earmark-binary-fill",
    "class": "bi-file-earmark-binary-fill",
    "c": "bi-file-earmark-code-fill",
    "go": "bi-file-earmark-code-fill",
    "cpp": "bi-file-earmark-code-fill",
    "zip": "bi-file-earmark-zip-fill",
    "jar": "bi-file-earmark-zip-fill",
    "rar": "bi-file-earmark-zip-fill",
    "tar": "bi-file-earmark-zip-fill",
    "ppt": "bi-file-earmark-slides-fill",
    "xls": "bi-file-earmark-spreadsheet-fill",
    "xlsx": "bi-file-earmark-spreadsheet-fill",
    "csv": "bi-file-earmark-spreadsheet-fill",
    "mp3": "bi-file-earmark-music-fill",
    "wav": "bi-file-earmark-music-fill",
    "other": "bi-file-earmark-fill"
}

```

```

static uploadFile(){
    var ele=document.createElement('input')
    ele.type='file'
    ele.setAttribute("multiple","")
    ele.click()
    ele.addEventListener("change",()=>{
        var arr=ele.files;
        for(let i of arr){
            FileManager.uploadFile(i)
        }
    })
}

```

```

static getExt(name){
    var arr=name.split(".")
    return arr[arr.length-1]
}

static addFile(id,name){
    var ext=this.getExt(name)
    if(!(ext in this.ext_map)) ext="other"
    var ele=document.createElement("div")
    ele.id=id+"_file"
    ele.setAttribute("name",name)
    ele.setAttribute("f_type","file")
    ele.classList.add("f_cont")
}

```

```

        ele.addEventListener("click",()=>{
            this.setActive(ele)
        })
        ele.innerHTML=`<div class="f_inner_cont">
            <div class="f_icon"><i class="bi `+this.ext_map[ext]+`"></i></div>
            <div class="f_name">`+name+`</div>
        </div>`
        document.getElementById("file_explorer").append(ele)
        return ele;
    }
    static addFolder(id,name){
        var ele=document.createElement("div")
        ele.id=id+"_folder"
        ele.setAttribute("name",name)
        ele.setAttribute("f_type","folder")
        ele.classList.add("f_cont")
        ele.addEventListener("click",()=>{
            this.setActive(ele)
        })
        ele.addEventListener("dblclick",()=>{
            storage.loadFolder(storage.getFolderById(id,storage.storage))
        })
        ele.innerHTML=`<div class="f_inner_cont">
            <div class="f_icon"><i class="bi bi-folder-fill"></i></div>
            <div class="f_name">`+name+`</div>
        </div>`
        document.getElementById("file_explorer").append(ele)
        return ele;
    }
    static removeFile(data){
        var ele=data
        if(typeof data === 'string' || typeof data === 'number'){
            ele=document.getElementById(data+"_file")
        }
        ele.remove()
    }
    static removeFolder(data){
        var ele=data
        if(typeof data === 'string' || typeof data === 'number'){
            ele=document.getElementById(data+"_folder")
        }
        ele.remove()
    }
    static clear(){
        document.getElementById("file_explorer").innerHTML=""
    }

```

```

        document.getElementById("menu_selected").innerHTML=""
    }
    static setActive(ele){
        var id=ele.id.split('_')[0];
        var root=document.getElementsByClassName("f_selected")
        for(let i=0;i<root.length;i++){
            root[i].classList.remove("f_selected")
        }
        ele.classList.toggle("f_selected")
        if(ele.getAttribute("f_type")== 'folder'){
            document.getElementById("menu_selected").innerHTML=
            `<div class="menu" onclick="storage.loadFolder(storage.getFolderById(`+id
+`,storage.storage))">
                <i class="bi bi-folder-symlink"></i>
                <span>Open Folder</span>
            </div>
            <div class="menu" onclick='Shower.show_folder_renamer(`+id+`,`+ele.getAt
tribute('name')+`")'>
                <i class="bi bi-pencil-square"></i>
                <span>Rename Folder</span>
            </div>
            <div class="menu" onclick='FolderManager.deleteFolder(`+id+`)'>
                <i class="bi bi-trash"></i>
                <span>Delete Folder</span>
            </div>
            <div class="menu" onclick='Shower.show_folder_details("`+ele.getAttribute
('name')+`,`+id+`)'>
                <i class="bi bi-info-circle"></i>
                <span>Details</span>
            </div>`
        }
        else{
            document.getElementById("menu_selected").innerHTML=`<div class="menu"
onclick='FileManager.openFile(`+id+`)'>
                <i class="bi bi-file-earmark"></i>
                <span>Open File</span>
            </div>
            <div class="menu" onclick='Shower.show_file_renamer(`+id+`,`+ele.getAttr
ibute('name')+`")'>
                <i class="bi bi-pencil-square"></i>
                <span>Rename File</span>
            </div>
            <div class="menu" onclick='FileManager.deleteFile(`+id+`)'>
                <i class="bi bi-trash"></i>
                <span>Delete File</span>
            </div>`
        }
    }
}

```

```

        </div>
        <div class="menu" onclick='FileManager.downloadFile(`+id+`)'>
            <i class="bi bi-download"></i>
            <span>Download File</span>
        </div>
        <div class="menu" onclick='FileManager.copyFileUrl(`+id+`)'>
            <i class="bi bi-link"></i>
            <span>Copy Link</span>
        </div>
        <div class="menu" onclick='Shower.show_file_details(`+id+`,`+ele.getAttribute('name')+`)'>
            <i class="bi bi-info-circle"></i>
            <span>Details</span>
        </div>
    }
}
}

```

```

class FileLoader{
    storage={}
    currentFolder={};
    loadedIds=[]
    constructor(root){
        this.storage.id=root
        this.storage.name="home"
        this.storage.folders=[]
        this.storage.files=[]
        document.body.style.cursor='wait'
        $.ajax({
            type: "post",
            url: "ServerPage",
            data: {"type":"FOLDER INFO","id":root},
            success: (json)=>{
                document.body.style.cursor='default'
                this.storage.folders=json.folders;
                this.storage.files=json.files;
                this.loadedIds.push(root);
                this.loadFolder(this.storage);
                PathLoader.addHome();
            },
            error: (err)=>{
                document.body.style.cursor='default';
                console.log(err);
                alert("server error fileloader constructor");
            }
        })
    }
}

```

```

    })
  }
  getFileById(id, folder){
    var arr=folder.files;
    for(let i=0;i<arr.length;i++){
      if(arr[i].id==id) return arr[i];
    }
    arr=folder.folders;
    for(let i=0;i<arr.length;i++){
      var fol= this.getFileById(id,arr[i])
      if(fol!=null) return fol;
    }
    return null;
  }
  getFolderById(id, folder){
    if(folder.id==id) return folder;

    var arr=folder.folders;

    for(let i=0;i<arr.length;i++){
      var fol= this.getFolderById(id,arr[i])
      if(fol!=null) return fol;
    }
    return null;
  }
  loadFolder(folder){
    if(!this.loadedIds.includes(folder.id)) {
      this.setFolder(folder,()=>{this.loadFolder(folder)})
    }
    else{
      this.currentFolder=folder
      CloudFile.clear()
      if(this.storage!=folder) PathLoader.addFolder(folder.id, folder.name)
      var arr=folder.folders;
      for(let i=0;i<arr.length;i++){
        CloudFile.addFolder(arr[i].id, arr[i].name)
      }
      arr=folder.files;
      for(let i=0;i<arr.length;i++){
        CloudFile.addFile(arr[i].id, arr[i].name)
      }
    }
  }
  setFolder(folder, then){
    var id=folder.id

```

```

        document.body.style.cursor='wait'
        $.ajax({
            type: "POST",
            url: "ServerPage",
            data: {"type":"FOLDER INFO","id":id},

            success: (json)=>{
                document.body.style.cursor='default'
                folder.folders=json.folders;
                folder.files=json.files;
                this.loadedIds.push(id)
                if(then!=undefined) then()
            },
            error: (e)=>{
                document.body.style.cursor='default'
                alert("server error");
            }
        })
    }
}

class FolderManager{
    static loadFolderSize(id){
        document.body.style.cursor='wait'
        $.ajax({
            type: "post",
            url: "ServerPage",
            data: {"type":"FOLDER SIZE","id":id},

            success: (json)=>{
                document.body.style.cursor='default'
                $('#.value')[3].innerText=FileManager.getRelativeSize(json.size)
            },
            error: ()=>{
                document.body.style.cursor='default'
                alert("server error");
            }
        })
    }
}

static loadStorage(){
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: { "type" : "FOLDER SIZE" , "id" : storage.storage.id },
    })
}

```

```

        success: (json)=>{
            document.body.style.cursor='default'
            var p=json.size/(2*1024*1024*10);
            $('#storage_value')[0].style.width=p+"%";
            $('#relative_storage').text(FileManager.getRelativeSize(json.size
)+" OF 2 GB");
        },
        error: ()=>{
            document.body.style.cursor='default'
            alert("server error");
        }
    })
}

static createFolder(){
    var name=$('#front input').val()
    var parent=storage.currentFolder.id
    if(name.length<1) {return;}
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FOLDER CREATE","name":name,"parent":parent},

        success: (json)=>{
            document.body.style.cursor='default'

            storage.currentFolder.folders.push(json.folder)
            storage.loadFolder(storage.currentFolder)

            Shower.remove_back()
        },
        error: ()=>{
            document.body.style.cursor='default'
            alert("server error");
        }
    })
}

static renameFolder(name,id){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FOLDER RENAME","name":name,"id":id},

        success: (json)=>{

```

```

        document.body.style.cursor='default'

        storage.getFolderById(id,storage.storage).name=name
        storage.loadFolder(storage.currentFolder)
        Shower.remove_back()

    },
    error: ()=>{
        document.body.style.cursor='default'
        alert("server error");
        Shower.remove_back()
    }
})
}
static deleteFolder(id){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FOLDER DELETE","id":id},

        success: (json)=>{
            document.body.style.cursor='default'

            var f=storage.getFolderById(id,storage.storage)
            storage.currentFolder.folders.splice(storage.currentFolder.folders.indexOf(f),1)
            storage.loadFolder(storage.currentFolder)

        },
        error: ()=>{
            document.body.style.cursor='default'
            alert("server error");
        }
    })
}
static putFolderDetails(id){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FOLDER SIZE","id":id},

        success: (json)=>{
            document.body.style.cursor='default'

```



```

        ele.innerText=json.size
    },
    error: ()=>{
        document.body.style.cursor='default'
        alert("server error");
    }
})
}
}

class FileManager{
    static getRelativeSize(bytes){
        if(bytes<1024) return bytes+" bytes"
        else if(bytes/1024 < 1024) return (bytes/1024).toFixed(0) + " KB"
        else if(bytes/(1024*1024) < 1024) return (bytes/(1024*1024)).toFixed(1) +
" MB"
        else if(bytes/(1024*1024*1024) < 1024) return (bytes/(1024*1024*1024)).to
Fixed(2) + " GB"
        else return "unknown"
    }
    static uploadFile(ufile){
        document.body.style.cursor='wait'
        var fd=new FormData()
        fd.append("file",ufile)
        fd.append("type","FILE UPLOAD")
        fd.append("parent",storage.currentFolder.id)
        $.ajax({
            type: "post",
            url: "ServerPage",
            processData: false,
            contentType: false,
            data: fd,
            success: (json)=>{
                document.body.style.cursor='default'
                if(json.status){
                    storage.currentFolder.files.push(json.file)
                    storage.loadFolder(storage.currentFolder)
                    FolderManager.loadStorage()
                }
                else{
                    alert("storage full");
                }
            }
        },

```

```

        error: (err)=>{
            document.body.style.cursor='default'
            var json=err.responseText;
            if(json.status){

                storage.currentFolder.files.push(json.file)
                storage.loadFolder(storage.currentFolder)
            }
            else{
                alert("storage full");
            }
        }
    })
}

static loadInfos(id,name){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FILE INFO","id":id,"name":name},

        success: (json)=>{
            document.body.style.cursor='default'

            $('#.value')[2].innerText=FileManager.getRelativeSize(json.size)

            $('#.value')[3].innerText=json.lastmodified

        },
        error: ()=>{
            document.body.style.cursor='default'
            alert("server error");
        }
    })
}

static deleteFile(id){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FILE DELETE","id":id},

        success: (json)=>{
            document.body.style.cursor='default'

```

```

        var f=storage.getFileById(id,storage.storage)
        storage.currentFolder.files.splice(storage.currentFolder.files.indexOf(f),1)

        storage.loadFolder(storage.currentFolder)
        FolderManager.loadStorage();
    },
    error: ()=>{
        document.body.style.cursor='default'
        alert("server error");
    }
})
}

static getFileUrl(id,download){
    var url=new URL("http://localhost:7060/fileurl");
    url.searchParams.append('id',id)
    if(download) url.searchParams.append('download',true)
    return url.href
}

static copyFileUrl(id){
    navigator.clipboard.writeText(this.getFileUrl(id))
}

static downloadFile(id){
    window.open(this.getFileUrl(id,true))
}

static openFile(id){
    window.open(this.getFileUrl(id))
}

static renameFile(name,id){
    document.body.style.cursor='wait'
    $.ajax({
        type: "post",
        url: "ServerPage",
        data: {"type":"FILE RENAME","id":id,"name":name},
        success: (json)=>{
            document.body.style.cursor='default'
            console.log(json.status)
            storage.getFileById(id,storage.storage).name=name
            storage.loadFolder(storage.currentFolder)
            Shower.remove_back()
        },
        error: ()=>{
            document.body.style.cursor='default'
            alert("server error");
            Shower.remove_back();
        }
    })
}

```

```

    }
  })
}

class PathLoader{
  static addHome(){
    document.getElementById("path_cont").innerHTML='<div id="home" class="folder_path"><i class="bi bi-house"></i><span>Home</span></div>'
    document.getElementById("home").addEventListener("click",()=>{
      storage.loadFolder(storage.storage)
      this.addHome()
    })
  }
  static addFolder(id,name){
    if($("#"+id+"_path").length!=0) return;
    var ele=document.createElement("div")
    ele.classList.add("folder_path")
    ele.id=id+"_path"
    ele.addEventListener("click",()=>{
      this.remove(id)
      storage.loadFolder(storage.getFolderById(id,storage.storage))
    })
    ele.innerHTML='<span>'+name+'</span>'
    document.getElementById("path_cont").append(ele)
  }
  static remove(id){
    var ele= document.getElementById(id+"_path")
    for(let i=document.getElementById("path_cont").childElementCount-1;i>=0;i--){
      if(document.getElementById("path_cont").children[i]==ele) break;
      document.getElementById("path_cont").children[i].remove()
    }
    ele.remove()
  }
  static getPath(){
    return document.getElementById("path_cont").innerText.replaceAll('\n','\\
');
  }
}

window.getCookie = function(name) {
  var match = document.cookie.match(new RegExp('(\\| )' + name + '=(\\[\\;]+)'));
  if (match) return match[2];
}

```

```

document.addEventListener("DOMContentLoaded",()=>{
    document.body.style.cursor='default';
    document.getElementById("file_explorer").addEventListener("dragenter",(evt)
=>{
        evt.preventDefault();
    })
    document.getElementById("file_explorer").addEventListener("dragover",(evt)
=>{
        evt.preventDefault();
    })
    document.getElementById("file_explorer").addEventListener("drop",(e)=>{
        var files = (e.dataTransfer.files);
        for(let i of files){
            FileManager.uploadFile(i)
        }
        window.files=files
        e.preventDefault();    })
})

```

JAVA FILES:

Jcloud (Package)

ServerPage.java:

```

package jcloud;

import dbms.*;
import java.io.*;
import java.math.BigInteger;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet(name = "ServerPage", urlPatterns = {"/ServerPage"})

@MultipartConfig(
    fileSizeThreshold = 1024 * 1024 * 1,

```

```

    maxFileSize = 1024 * 1024 * 20,
    maxRequestSize = 1024 * 1024 * 100
)

public class ServerPage extends HttpServlet {

    public static String pppath="/home/kowsik/Documents/Netbeans/jcloud_linux/src/main/java/data/profile_pic/";
    public static String fpath="/home/kowsik/Documents/Netbeans/jcloud_linux/src/main/java/data/files/";

    public static UserTable ut=new UserTable();
    public static FolderTable fot=new FolderTable();
    public static FileTable fit=new FileTable();

    public static OTPManager mm=new OTPManager();

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        if(req.getParameter("type")==null) return;

        if(req.getParameter("type").equals("SIGNIN")){

            String email=req.getParameter("email");
            String password=req.getParameter("password");

            // dbms process

            String status=ut.signIn(email,password);
            if(status.equals("SIGNIN SUCCESS")){
                resp.addCookie(new Cookie("email",email));
                resp.addCookie(new Cookie("password",password));
            }

            // json response

            JSONWriter out=new JSONWriter(resp);
            out.setObjectValue("status", status);
            out.close();
        }
        else if(req.getParameter("type").contains("SIGNUP")){

            String email=req.getParameter("email");
            String password=req.getParameter("password");

```

```

        // profile picture upload

        Part p=req.getPart("img");
        File f=new File(pppath+email);
        if(!f.exists()) f.mkdir();
        InputStream pi=p.getInputStream();
        FileOutputStream foss=new FileOutputStream(pppath+email+"/"+p.getSubmittedFileName() );
        byte[] buffer = new byte[4096];
        int length;
        while((length = pi.read(buffer)) > 0) foss.write(buffer, 0, length);

        // dbms process

        String status=ut.signUp(req.getParameter("fullname"),email,password,"");

        if(status.equals("SIGNUP SUCCESS")){
            resp.addCookie(new Cookie("email",email));
            resp.addCookie(new Cookie("password",password));
        }

        // json response

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", status);
        out.close();
    }

    else if(req.getParameter("type").contains("PROFILE INFO")){
        if(auth(req)){ // authentication from cookies

            // writing json response

            JSONWriter out=new JSONWriter(resp);
            out.setObjectValue("status", true);
            out.setObjectValue("fullname", ut.getName(req));
            out.setObjectValue("storage", ut.getStorage(req));
            out.setObjectValue("root",ut.getRootId(req));
            out.close();

        }
    }

    else if(req.getParameter("type").contains("FORGOT_PASSWORD RESET")){

```

```

        JSONWriter out=new JSONWriter(resp);
        System.out.println(req.getParameter("otp")+" "+req.getParameter("password"));

        // checking otp
        if(mm.checkOTP(Dbms.getCookieValue("email",req), req.getParameter("otp")) {

            String status=ut.updatePassword(req, req.getParameter("password"));

            if(status.equals("FORGOT_PASSWORD SUCCESS")) {
                resp.addCookie(new Cookie("password",req.getParameter("password")));
                mm.deleteEntry(Dbms.getCookieValue("email",req));
            }

            out.setObjectValue("status", status);
        }
        else
            out.setObjectValue("status", "FORGOT_PASSWORD WRONG_OTP");

        out.close();
    }

    else if(req.getParameter("type").contains("FORGOT_PASSWORD SEND_OTP")){

        JSONWriter out=new JSONWriter(resp);
        String status="FORGOT_PASSWORD FAILED";

        // sending otp to user mail id
        if(ut.isEmailThere(req.getParameter("email"))){
            if(mm.sendOTP(req.getParameter("email"))){
                status="FORGOT_PASSWORD OTP_SENT";
                resp.addCookie(new Cookie("email",req.getParameter("email")))
            }
        }
        else status="FORGOT_PASSWORD EMAIL_NOT_FOUND";
        out.setObjectValue("status", status);
        out.close();
    }

    else if(req.getParameter("type").contains("UPDATE FULLNAME")){

```



```

        if(auth(req)){
            // writing json response
            JSONWriter out=new JSONWriter(resp);
            out.setObjectValue("status", ut.updateName(req, req.getParameter(
"name"))));
            out.close();
        }
    }

    else if(req.getParameter("type").contains("FOLDER INFO")){
        if(auth(req)){
            int id=Integer.parseInt(req.getParameter("id"));
            String name=fot.getName(id);

            // file and folder list inside given folder
            ArrayList<SqlFile> childfiles=fit.getFileList(id);
            ArrayList<SqlFile> childfolders=fot.getFolderList(id);

            // writing json response
            JSONWriter out=new JSONWriter(resp);
            out.setObjectValue("name", name);
            out.setObjectValue("id", id);
            out.setObjectValue("folders", childfolders);
            out.setObjectValue("files", childfiles);
            out.close();
        }
    }

    else if(req.getParameter("type").contains("FILE UPLOAD")){
        if(auth(req)){

            JSONWriter out=new JSONWriter(resp);
            Part p=req.getPart("file");

            // ensure storage space
            BigInteger max = new BigInteger("2147483648");
            BigInteger curr = new BigInteger((ut.getStorage(req)+p.getSize())
+""));

            if( max.compareTo(curr)!=-1 ){
                String fileName=p.getSubmittedFileName();

```

```

        int id=fit.createFile(p.getSubmittedFileName(),getExt(fileName), Integer.parseInt(req.getParameter("parent")),p.getSize(),req);
        if(id!=-1){
            p.write(fpath+ id+"."+getExt(fileName));
            out.setObjectValue("status", true);
            out.setObjectValue("file",SqlFile.getFile(fileName, id) );
        }
        else out.setObjectValue("status", false);
    }

    else out.setObjectValue("status", false);
    out.close();

}

}

else if(req.getParameter("type").contains("FILE RENAME")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", fit.renameFile(Integer.parseInt(req.getParameter("id")),req.getParameter("name") ));
        out.close();

    }

}

else if(req.getParameter("type").contains("FILE DELETE")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", fit.deleteFile(Integer.parseInt(req.getParameter("id")), req));
        out.close();

    }

}

else if(req.getParameter("type").contains("FILE INFO")){
    if(auth(req)){
        // writing size and last modified infos
        JSONWriter out=new JSONWriter(resp);

```

```

        File f=new File(fpath+req.getParameter("id")+ "."+getExt(req.getPa
parameter("name")));
        out.setObjectValue("status", true);
        out.setObjectValue("size", f.length());

        SimpleDateFormat sdf = new SimpleDateFormat("MM-dd-
yyyy HH:mm:ss");
        out.setObjectValue("lastmodified", sdf.format(f.lastModified()) )
;

        out.close();

    }
}

else if(req.getParameter("type").contains("FOLDER CREATE")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        int id=fot.createFolder(req.getParameter("name"), Integer.parseIn
t(req.getParameter("parent")));
        out.setObjectValue("status", (id!=-1) );
        out.setObjectValue("folder", SqlFile.getFolder(req.getParameter("
name"), id));
        out.close();

    }
}

else if(req.getParameter("type").contains("FOLDER DELETE")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", fot.deleteFolder(Integer.parseInt(re
q.getParameter("id")), req));
        out.close();

    }
}

else if(req.getParameter("type").contains("FOLDER RENAME")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", fot.renameFolder(req.getParameter("n
ame"),Integer.parseInt( req.getParameter("id"))));

```

```

        out.close();

    }
}
else if(req.getParameter("type").contains("FOLDER SIZE")){
    if(auth(req)){

        JSONWriter out=new JSONWriter(resp);
        out.setObjectValue("status", true);
        out.setObjectValue("size", fot.getFolderSize(Integer.parseInt(req
.getParameter("id"))));
        out.close();

    }
}

}

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
    if(req.getParameter("type")==null) return;

    if(req.getParameter("type").contains("PROFILE PICTURE")){
        if(auth(req)){

            // profile picture response

            File f=new File(pppath+Dbms.getCookieValue("email", req)).listFil
es()[0];

            ServletContext cntx= req.getServletContext();
            String mime = cntx.getMimeType(f.getName());
            if (mime == null) return;

            resp.setContentType(mime);
            resp.setHeader("Content-
disposition","inline; filename="+f.getName());

            FileInputStream fis= new FileInputStream(f);
            OutputStream out = resp.getOutputStream();
            byte[] buffer = new byte[4096];
            int length;
            while ((length = fis.read(buffer)) > 0) out.write(buffer, 0, leng
th);

            fis.close();
            out.flush();

```

```

    }
}

public static boolean auth(HttpServletRequest req){
    String email=null;
    String password=null;
    Cookie[] arr= req.getCookies();
    if(arr==null) return false;
    for(Cookie i: arr){
        if(i.getName().equals("email")) email=i.getValue();
        else if(i.getName().equals("password")) password=i.getValue();
    }
    if(ut.signIn(email,password).equals("SIGNIN SUCCESS")) return true;
    return false;
}

public static String getExt(String filename){
    String[] arr=filename.split("[.]");
    return arr[arr.length-1];
}
}

```

OTPManager.java:

```

package jcloud;

import java.util.HashMap;

public class OTPManager {
    HashMap<String,String> otps=new HashMap<String,String>();
    public boolean checkOTP(String email,String otp){
        return otps.get(email).equals(otp);
    }
    public void deleteEntry(String email){
        otps.remove(email);
    }
    private String generateOTP(){
        int otp1=(int)(Math.random()*10);
        int otp2=(int)(Math.random()*10);
        int otp3=(int)(Math.random()*10);
        int otp4=(int)(Math.random()*10);
        return otp1+""+otp2+""+otp3+""+otp4;
    }
}

```

```

    }
    public boolean sendOTP(String email){
        String otp=generateOTP();
        otps.put(email, otp);
        String html="<body style=\"margin: 0; height: 520px; background-
color: rgb(255, 255, 255);\">\r\n"
            + "    <link rel=\"preconnect\" href=\"https://fonts.googleap
is.com\">\r\n"
            + "    <link rel=\"preconnect\" href=\"https://fonts.gstatic.
com\" crossorigin>\r\n"
            + "    <link href=\"https://fonts.googleapis.com/css2?family=
Open+Sans:wght@600&display=swap\" rel=\"stylesheet\">\r\n"
            + "    <div id=\"holder\" style=\"margin: 10px;height: 300px;
width: calc(100% - 20px);background-color: rgb(255, 255, 255);margin-
top: 75px;\">\r\n"
            + "        <div id=\"box\" style=\"padding-
top: 5px; background-color: rgb(238, 238, 255);height: 150px;width: 300px;border-
radius: 10px;margin: auto;\">\r\n"
            + "            <div id=\"title\" style=\"width: 100%;font-
size: x-large;font-family: 'Open Sans', sans-serif;text-align: center;margin-
top: 20px;color: rgb(128, 20, 77);\">verification code</div>\r\n"
            + "            <div id=\"otp\" style=\"width: fit-
content;margin: auto; margin-top: 14px;\">\r\n"
            + "                <div style=\"height: 30px;width: 25px;bord
er-radius: 5px;background-color: rgb(221, 221, 221);margin: 5px;text-
align: center;font-size: x-
large;color: rgb(73, 73, 73);float: left;\">"+otp.charAt(0)+"</div>\r\n"
            + "                <div style=\"height: 30px;width: 25px;bord
er-radius: 5px;background-color: rgb(221, 221, 221);margin: 5px;text-
align: center;font-size: x-
large;color: rgb(73, 73, 73);float: left;\">"+otp.charAt(1)+"</div>\r\n"
            + "                <div style=\"height: 30px;width: 25px;bord
er-radius: 5px;background-color: rgb(221, 221, 221);margin: 5px;text-
align: center;font-size: x-
large;color: rgb(73, 73, 73);float: left;\">"+otp.charAt(2)+"</div>\r\n"
            + "                <div style=\"height: 30px;width: 25px;bord
er-radius: 5px;background-color: rgb(221, 221, 221);margin: 5px;text-
align: center;font-size: x-
large;color: rgb(73, 73, 73);float: left;\">"+otp.charAt(3)+"</div>\r\n"
            + "            </div>\r\n"
            + "        </div>\r\n"
            + "    </div>\r\n"
            + "</body>";
        return MailSender.sendMail(html, email);
    }

```

```

        public static void main(String[] args) {
            System.out.println(new OTPManager().generateOTP());
        }
    }
}

```

FileGetter.java

```

package jcloud;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.Arrays;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet(name = "FileGetter", urlPatterns = {"/fileurl"})
public class FileGetter extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
        ServletException, IOException {
        if(ServerPage.auth(req)){
            File f=new File(ServerPage.fpath+req.getParameter("id")+ "." +ServerPage
e.fit.getExt(Integer.parseInt(req.getParameter("id"))));
            String respType="";
            if(req.getParameter("download")!=null) respType="attachment; ";
            else respType="inline; ";

            resp.setHeader("Content-
disposition",respType+"filename="+f.getName());

            FileInputStream fis= new FileInputStream(f);

            OutputStream out = resp.getOutputStream();
            byte[] buffer = new byte[4096];
            int length;
            while ((length = fis.read(buffer)) > 0){
                out.write(buffer, 0, length);
            }
        }
    }
}

```

```

        fis.close();
        out.flush();
    }
}
}

```

JSONWriter.java:

```

package jcloud;

import dbms.SqlFile;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.HashMap;
import javax.servlet.http.HttpServletResponse;

public class JSONWriter{
    PrintWriter out=null;
    HashMap<String, String> map=null;
    public JSONWriter(HttpServletResponse response) {
        try {
            response.setContentType("text/json");
            out=response.getWriter();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public void setObjectValue(String key, Object value){
        if(map==null) map=new HashMap<String,String>();
        if(value instanceof String) map.put("\""+key+"\"", "\"" +value.toString()+
"\"");
        else map.put("\""+key+"\"", value.toString());
    }
    public void setContent(Object value){
        out.print(value.toString());
    }
    public void close(){
        if(map!=null) out.print(map.toString().replaceAll("=", ":") );
        out.close();
    }
}

```


MailSender.java:

```
package jcloud;

import java.util.Date;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.mail.*;
import javax.mail.internet.*;

public class MailSender {

    public static void sendPlainTextEmail(String host, String port,
        final String userName, final String password, String toAddress,
        String subject, String message) throws AddressException,
        MessagingException {

        // sets SMTP server properties
        Properties properties = new Properties();
        properties.put("mail.smtp.host", host);
        properties.put("mail.smtp.port", port);
        properties.put("mail.smtp.auth", "true");
        properties.put("mail.smtp.starttls.enable", "true");
        // *** BEGIN CHANGE
        properties.put("mail.smtp.user", userName);

        // creates a new session, no Authenticator (will connect() Later)
        Session session = Session.getDefaultInstance(properties);
        // *** END CHANGE

        // creates a new e-mail message
        Message msg = new MimeMessage(session);

        msg.setFrom(new InternetAddress(userName));
        InternetAddress[] toAddresses = { new InternetAddress(toAddress) };
        msg.setRecipients(Message.RecipientType.TO, toAddresses);
        msg.setSubject(subject);
        msg.setSentDate(new Date());
        // set plain text message
        msg.setText(message);

        // *** BEGIN CHANGE
        // sends the e-mail
        Transport t = session.getTransport("smtp");
```

```

        t.connect(userName, password);
        t.sendMessage(msg, msg.getAllRecipients());
        t.close();
// *** END CHANGE

    }

    public static boolean sendMail(String htmlCode, String to) {

        Properties props = new Properties();
        System.out.println(to);

        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.socketFactory.port", "587");
        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "587");

        Authenticator a = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(EmailAddress.EMAIL, EmailAddress.PASSWORD);
            }
        };

        try {
            Session s = Session.getInstance(props, a);

            Message msg = new MimeMessage(s);

            msg.setFrom(new InternetAddress(EmailAddress.EMAIL));

            msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));

            msg.setSubject("Reset your password");

            msg.setText("fdvb");

            MimeMultipart multipart = new MimeMultipart("related");

            BodyPart messageBodyPart = new MimeBodyPart();
            String htmlText = htmlCode;

```

```

        messageBodyPart.setContent(htmlText, "text/html");
        multipart.addBodyPart(messageBodyPart);

        msg.setContent(multipart);

        Transport.send(msg);

        return true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

public static void main(String[] args){
    try {
        sendPlainTextEmail("smtp.gmail.com", "465", "kowsikravi123@gmail.com",
        , "8838163291", "kowsikravi112@gmail.com", "java test", "success");
    } catch (MessagingException ex) {
        Logger.getLogger(MailSender.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

Package: dbms

Dmbs.java:

```

package dbms;

import java.sql.*;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;

public class Dbms {

    Connection con;
    Statement st;

    public Dbms() {

```

```

        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cloud"
,"kowsik", "9940487477");
            con.setAutoCommit(true);
            st = con.createStatement();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }

    public boolean query(String s) {
        try {
            st.executeUpdate(s);
            return true;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public PreparedStatement getPs(String query) {
        try {
            return con.prepareStatement(query);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    public void close() {
        try {
            con.close();
        } catch (Exception e) {
        }
    }

    public static String getCookieValue(String key,HttpServletRequest req){
        Cookie[] arr= req.getCookies();
        if(arr==null) return null;
        for(Cookie i : arr){
            if(i.getName().equals(key)) return i.getValue();
        }
        return null;
    }
}

```

```

        public static void main(String[] args) throws Exception{
            System.out.println(new Dbms().getPs("select * from users").executeQuery()
.next());
        }
    }
}

```

UserTable.java

```

package dbms;

import java.io.File;
import java.math.BigDecimal;
import java.math.BigInteger;
import java.sql.*;
import javax.crypto.*;
import javax.crypto.spec.*;
import java.security.spec.*;
import java.util.*;
import javax.servlet.http.HttpServletRequest;

public class UserTable extends Dbms {

    public void createTable() {
        System.out.println(query("create table users("
            + "id int primary key auto_increment,"
            + "full_name varchar(100),"
            + "email varchar(200) unique,"
            + "pwd varchar(100),"
            + "root int,"
            + "storage_space int);"));
    }

    public void deleteTable() {
        System.out.println(query("drop table users;"));
    }

    public boolean isEmailThere(String email) {
        return (getUserId(email) != -1);
    }
}

```

```

    public String signUp(String fullname, String email, String password, String imgPath) {
        try {
            File img = new File(imgPath);
            if (img.length() > 15000000) {
                return "SIGNUP LARGE_IMAGE_SIZE";
            }
            if (getUserId(email) != -1) {
                return "SIGNUP EMAIL_EXISTS";
            }
            if (fullname.length() < 1 && password.length() < 1) {
                return "SIGNUP ZERO_LENGTH";
            }

            PreparedStatement ps = getPs("insert into users(full_name,email,pwd,root) values(?,?,?,?)");
            ps.setString(1, fullname);
            ps.setString(2, email);
            ps.setString(3, hashp(password));

            int rootid = new FolderTable().createFolder("root", -1);
            if (rootid == -1) {
                return "SIGNUP FAILED";
            }

            ps.setInt(4, rootid);
            ps.execute();
            return "SIGNUP SUCCESS";

        } catch (Exception e) {
            e.printStackTrace();
        }
        return "SIGNUP FAILED";
    }

    public String signIn(String email, String password) {

        try {
            PreparedStatement ps = getPs("select id from users where id=? and pwd=?");

            int userId = getUserId(email);
            if (userId == -1) {
                return "SIGNIN EMIAL_NOT_FOUND";
            }
            ps.setInt(1, userId);

```

```

        ps.setString(2, hashp(password));
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            return "SIGNIN SUCCESS";
        } else {
            return "SIGNIN INCORRECT_PASSWORD";
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "SIGNIN FAILED";
}

public String updateName(HttpServletRequest req, String newName) {
    try {
        if (newName.length() < 1) {
            return "UPDATE ZERO_LENGTH";
        }
        PreparedStatement ps = getPs("update users set full_name=? where email=?");
        ps.setString(1, newName);
        ps.setString(2, getCookieValue("email", req));
        if (ps.execute()) {
            return "UPDATE SUCCESS";
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "UPDATE FAILED";
}

public String updatePassword(HttpServletRequest req, String newPassword) {
    try {
        if (newPassword.length() < 1) {
            return "FORGOT_PASSWORD ZERO_LENGTH";
        }
        PreparedStatement ps = getPs("update users set pwd=? where email=?");

        ps.setString(1, hashp(newPassword));

        ps.setString(2, getCookieValue("email", req));
        ps.execute();
        return "FORGOT_PASSWORD SUCCESS";
    }
}

```

```

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "FORGOT_PASSWORD FAILED";
}

public boolean updateSize(long size, HttpServletRequest req) {
    try {
        PreparedStatement ps = getPs("update users set storage_space=? where email=?");
        ps.setLong(1, getStorage(req) + size);
        ps.setString(2, getCookieValue("email", req));
        return ps.execute();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

public String getName(HttpServletRequest req) {
    try {
        PreparedStatement ps = getPs("select full_name from users where email=?");
        ps.setString(1, getCookieValue("email", req));
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            return rs.getString("full_name");
        }
    } catch (Exception e) {
    }
    return null;
}

public long getStorage(HttpServletRequest req) {
    try {
        PreparedStatement ps = getPs("select storage_space from users where email=?");
        ps.setString(1, getCookieValue("email", req));
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            return rs.getLong("storage_space");
        }
    } catch (Exception e) {
    }
    return -1;
}

```



```

    }

    public float getStoragePercent(HttpServletRequest req) {
        long st = getStorage(req);
        BigDecimal max = new BigDecimal(st + "");
        float p = max.divide(new BigDecimal("2147483648")).multiply(new BigDecimal(
1("100))).floatValue();
        System.out.println(p);
        return p;
    }

    public int getRootId(HttpServletRequest req) {
        try {
            PreparedStatement ps = getPs("select root from users where email=?");
            ps.setString(1, getCookieValue("email", req));
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                return rs.getInt("root");
            }
        } catch (Exception e) {
        }
        return -1;
    }

    private int getUserId(String email) {
        try {
            PreparedStatement ps = getPs("select id from users where email=?");
            ps.setString(1, email);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                return rs.getInt("id");
            } else {
                return -1;
            }
        } catch (Exception e) {
            e.printStackTrace();
            return -1;
        }
    }

    private String hashp(String s) {
        byte[] salt = {95, 80, 32, 93, 22, -33, 45, -38, -73, 98, 60, -91, -
117, -11, 52, 29};
        KeySpec spec = new PBEKeySpec(s.toCharArray(), salt, 65536, 128);
        SecretKeyFactory f;

```

```

        byte[] hash = null;
        try {
            f = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
            hash = f.generateSecret(spec).getEncoded();
        } catch (Exception e) {
            e.printStackTrace();
        }
        Base64.Encoder enc = Base64.getEncoder();
        return enc.encodeToString(hash);
    }
}

```

FileManager.java:

```

package dbms;

import java.io.File;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import jcloud.ServerPage;
import static jcloud.ServerPage.fpath;
import static jcloud.ServerPage.getExt;

public class FileTable extends Dbms {
    public static UserTable ut=new UserTable();

    public void createTable() {
        System.out.println(query("create table files("
            + "id int primary key auto_increment,"
            + "fname varchar(100),"
            + "ext varchar(50),"
            + "parent int,"
            + "fsize int);"));
    }

    public void deleteTable() {
        System.out.println("drop table files");
    }
}

```

```

    public int createFile(String name, String ext, int parent, long size,HttpServletRequest req) {
        try {
            PreparedStatement ps = getPs("insert into files (fname,ext,parent,fsize) values(?,?,?,?)");
            ps.setString(1, name);
            ps.setString(2, ext);
            ps.setInt(3, parent);
            ps.setLong(4, size);
            ps.execute();
            PreparedStatement pss = getPs("select last_insert_id()");
            ResultSet rs = pss.executeQuery();
            if (rs.next()) {
                ut.updateSize(size, req);
                return rs.getInt("last_insert_id()");
            } else {
                return -1;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return -1;
    }

    public boolean deleteFile(int id,HttpServletRequest req){
        try{
            long size=getSize(id);

            File f=new File(ServerPage.fpath+id+"."+getExt(id));

            PreparedStatement ps=getPs("delete from files where id=?");
            ps.setInt(1, id);
            ut.updateSize(-size, req);
            ps.execute();
            return f.delete();
        }catch(Exception e){
            e.printStackTrace();
        }

        return false;
    }

    public boolean renameFile(int id,String newname){
        try {

```

```

        String ext=getExt(id);
        PreparedStatement ps=getPs("update files set fname=?,ext=? where id=?");

        ps.setString(1, newname);
        ps.setString(2, ServerPage.getExt(newname));
        ps.setInt(3, id);

        ps.execute();
        File f=new File(fpath+id+"."+ext);

        f.renameTo(new File(fpath+id+"."+getExt(id)) );

        return true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

public ArrayList<SqlFile> getFileList(int parentid) {
    try {
        ArrayList<SqlFile> sqlfile = new ArrayList<>();
        PreparedStatement ps = getPs("select id,fname from files where parent
        =?");

        ps.setInt(1, parentid);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            sqlfile.add(SqlFile.getFile(rs.getString("fname"), rs.getInt("id"
        )));
        }
        return sqlfile;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

public long getSize(int fid) {
    try {
        PreparedStatement ps = getPs("select fsize from files where id=?");
        ps.setInt(1, fid);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            return rs.getLong("fsize");
        }
    }
}

```

```

    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1;
}

public String getExt(int id){
    try{
        PreparedStatement ps=getPs("select ext from files where id=?");
        ps.setInt(1, id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) return rs.getString("ext");
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return null;
}
}

```

FolderManager.java:

```

package dbms;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;

public class FolderTable extends Dbms {

    public static FileTable fit = new FileTable();

    public void createTable() {
        System.out.println(query("create table folders("
            + "id int primary key auto_increment,"
            + "fname varchar(100),"
            + "parent int);"));
    }
}

```

```

public void deleteTable() {
    System.out.println(query("drop table folders;"));
}

public int createFolder(String fname, int parent) {
    try {
        PreparedStatement ps = getPs("insert into folders (fname,parent) values(?,?)");
        ps.setString(1, fname);
        ps.setInt(2, parent);
        ps.execute();
        PreparedStatement pss = getPs("select last_insert_id();");

        ResultSet rs = pss.executeQuery();
        if (rs.next()) {
            return rs.getInt("last_insert_id()");
        } else {
            return -1;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return -1;
}

public boolean deleteFolder(int fid, HttpServletRequest req) {
    if (new UserTable().getRootId(req) == fid) {
        return false;
    }
    try {
        PreparedStatement ps = getPs("delete from folders where id=?");
        ps.setInt(1, fid);
        if(ps.execute()) deleteInnerFolder(fid, req);

    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

public void deleteInnerFolder(int parentid,HttpServletRequest req){

    ArrayList<SqlFile> sf = getFolderList(parentid);
    ArrayList<SqlFile> sfi = fit.getFileList(parentid);

```

```

        for (SqlFile i : sfi) {
            fit.deleteFile(i.getId(), req);
        }
        if (sf.size() < 1) {
            return;
        }

        for (SqlFile i : sf) {
            deleteInnerFolder(i.getId(), req);
            try {
                PreparedStatement ps = getPs("delete from folders where id=?");
                ps.setInt(1, i.getId());
                ps.execute();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return;
    }

    public boolean renameFolder(String fname, int id) {
        try {
            PreparedStatement ps = getPs("update folders set fname=? where id=?");
;
            ps.setString(1, fname);
            ps.setInt(2, id);
            return ps.execute();

        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }

    public ArrayList<SqlFile> getFolderList(int parentid) {
        try {
            ArrayList<SqlFile> sqlfile = new ArrayList<>();
            PreparedStatement ps = getPs("select id,fname from folders where parent=?");
            ps.setInt(1, parentid);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {

```

```

        sqlfile.add(SqlFile.getFolder(rs.getString("fname"), rs.getInt("id"
d"))));
    }
    return sqlfile;
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

public long getFolderSize(int parentid) {
    ArrayList<SqlFile> sf = getFolderList(parentid);
    ArrayList<SqlFile> sfi = fit.getFileList(parentid);
    long totalsize = 0;
    for (SqlFile i : sfi) {
        totalsize += fit.getSize(i.getId());
    }
    if (sf.size() < 1) {return totalsize;}
    long Totsize = 0;
    for (SqlFile i : sf) {
        Totsize += getFolderSize(i.getId());
    }
    return totalsize + Totsize;
}

public String getName(int id){
    try {
        PreparedStatement ps=getPs("select fname from folders where id=?");
        ps.setInt(1,id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            return rs.getString("fname");
        }
        else{
            System.out.println("no records");
        }
    } catch (Exception e) {
        System.out.println("error:");
        e.printStackTrace();
    }
    return null;
}
}

```


SqlFile.java:

```
package dbms;

import java.io.File;
import jcloud.ServerPage;

public class SqlFile {
    private String name;
    private int id;
    private boolean flag;
    private SqlFile(){}
    public boolean isFile(){return !flag;}
    public boolean isFolder(){return flag;}
    public String getName(){return name;}
    public int getId(){return id;}
    public static SqlFile getFile(String name,int id){
        SqlFile fi=new SqlFile();
        fi.flag=false;
        fi.name=name;
        fi.id=id;
        return fi;
    }
    public static SqlFile getFolder(String name,int id){
        SqlFile fi=new SqlFile();
        fi.flag=true;
        fi.name=name;
        fi.id=id;
        return fi;
    }
    @Override
    public String toString(){
        if(isFolder()) return "{\"name\" : \""+name+"\", \"id\" : "+id+", \"folders\": [], \"files\": []}";
        else return "{\"name\" : \""+name+"\", \"id\" : "+id+"}";
    }
}
```