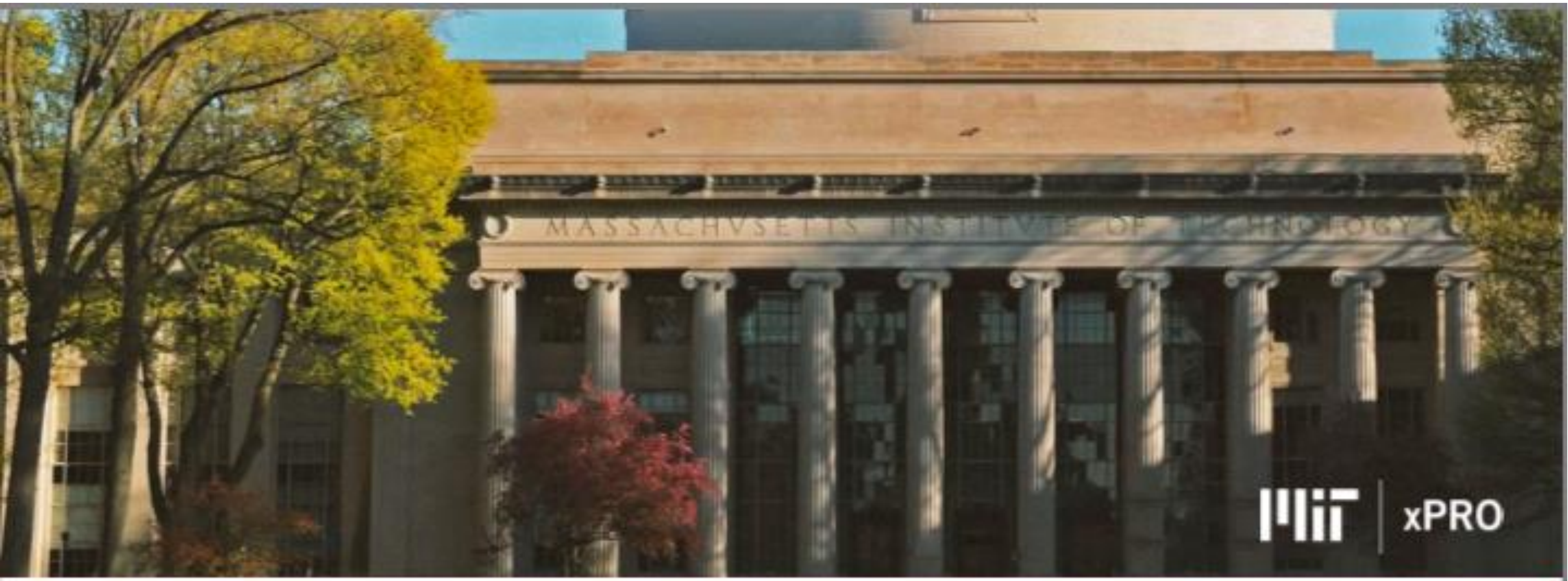
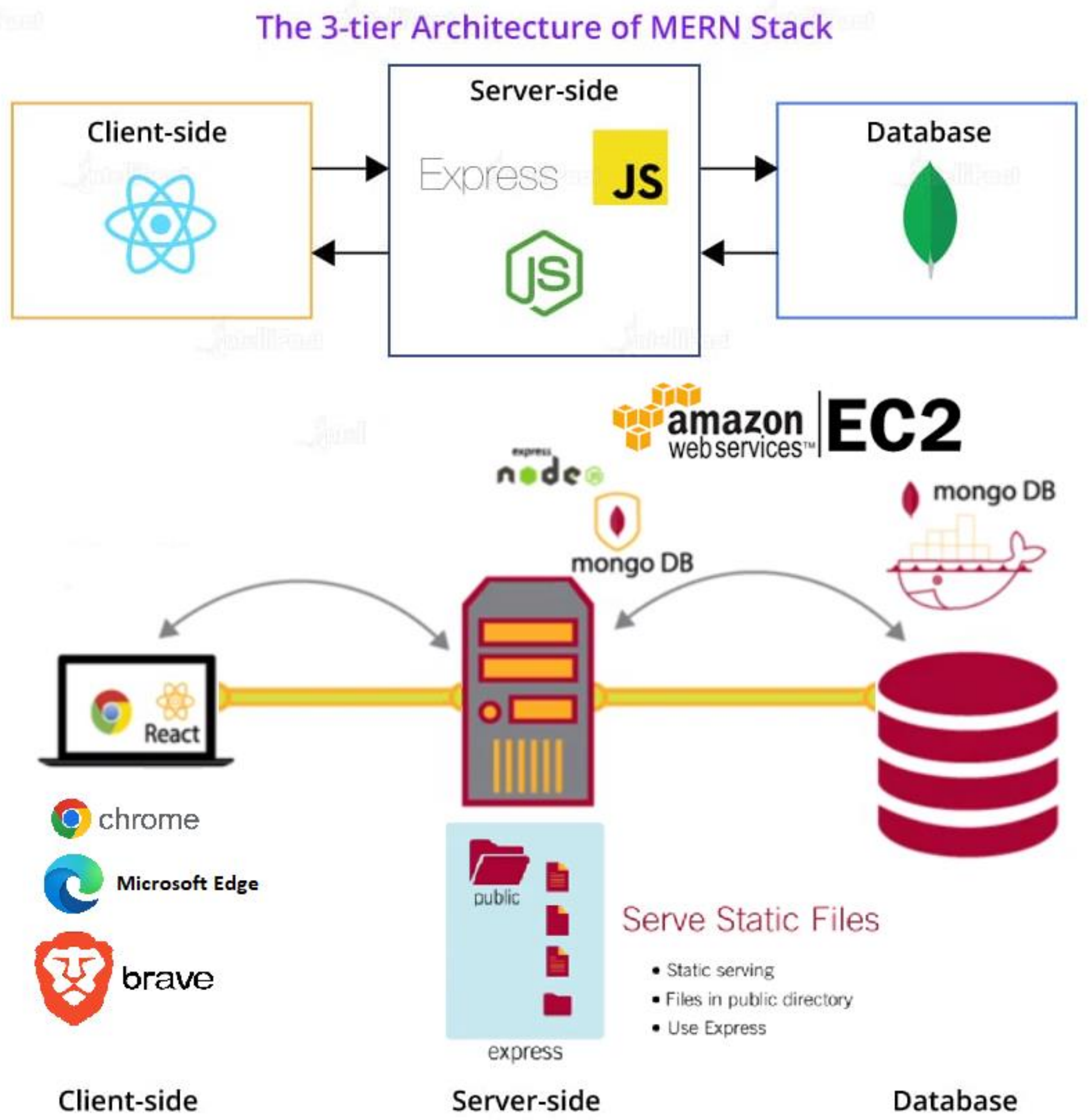


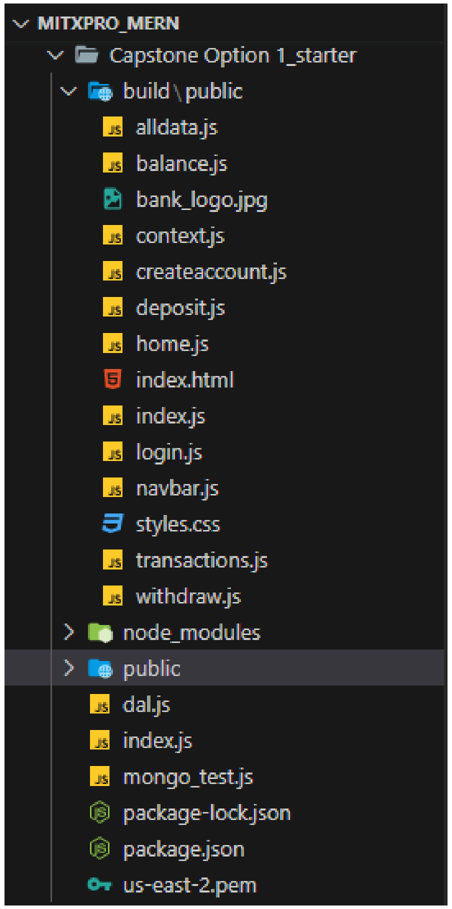
Banking Application Capstone Presentation

Omar Cerda Villalobos

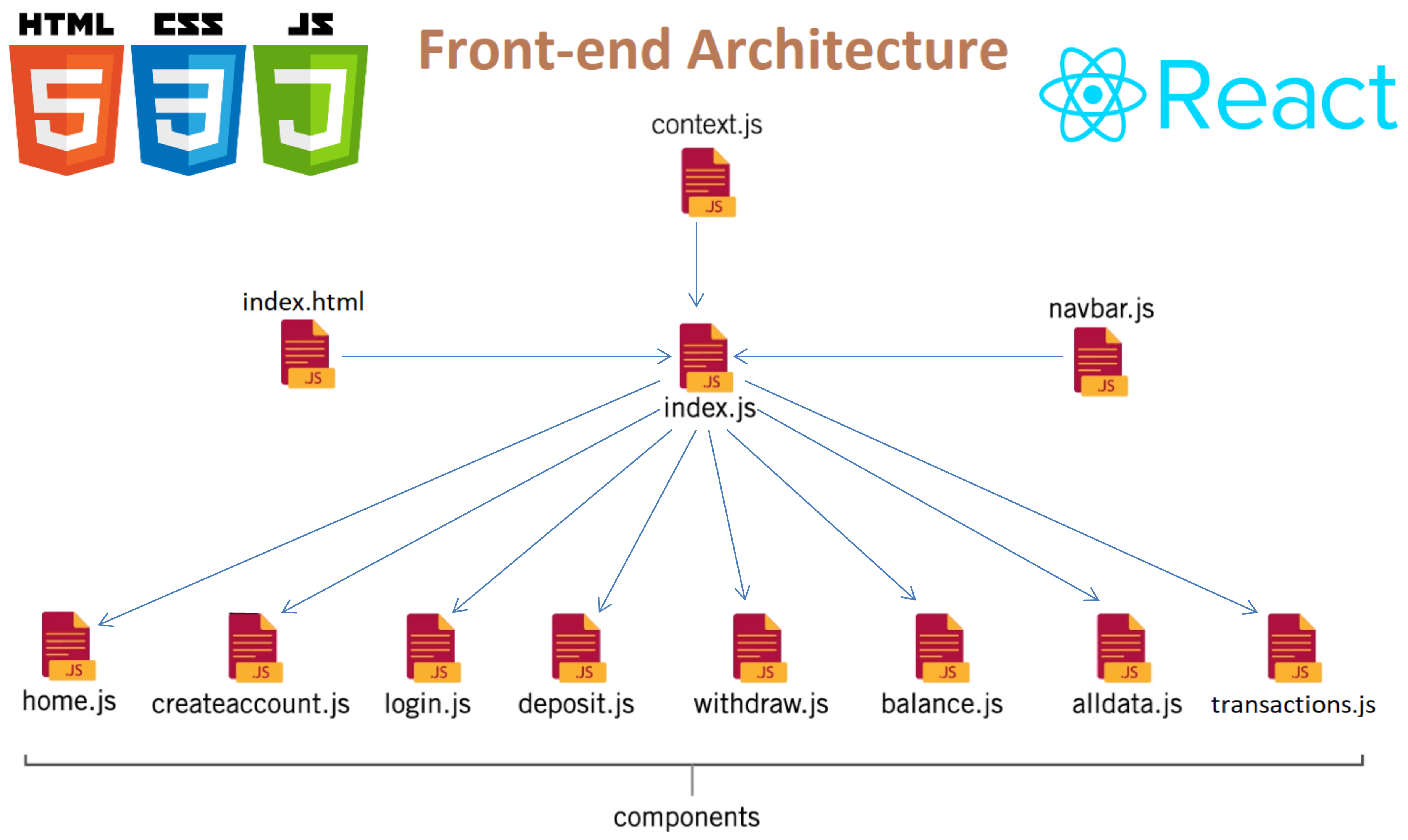
Part 1: Front-End Architecture, Authentication, And App Diagram

Application Overview Diagram

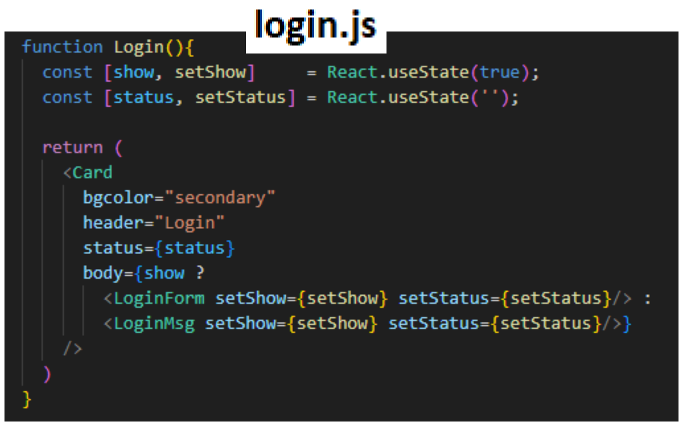



```
MITXPRO_MERN
  Capstone Option 1_starter
    build\public
      alldata.js
      balance.js
      bank_logo.jpg
      context.js
      createaccount.js
      deposit.js
      home.js
      index.html
      index.js
      login.js
      navbar.js
      styles.css
      transactions.js
      withdraw.js
    node_modules
    public
    dal.js
    index.js
    mongo_test.js
    package-lock.json
    package.json
    us-east-2.pem
```

Front-End Architecture

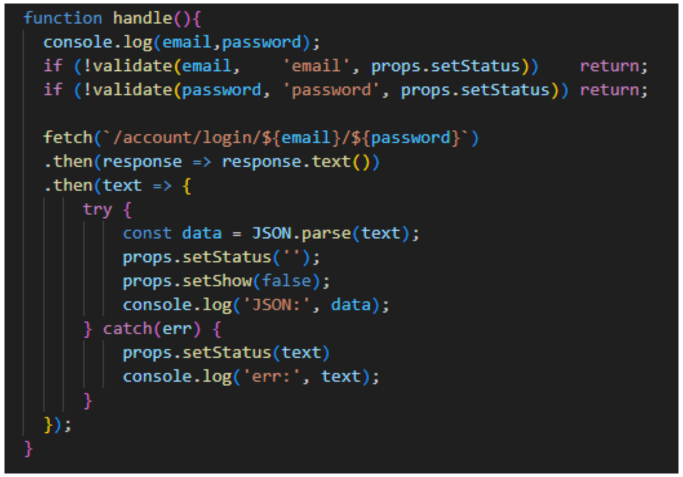


Authentication

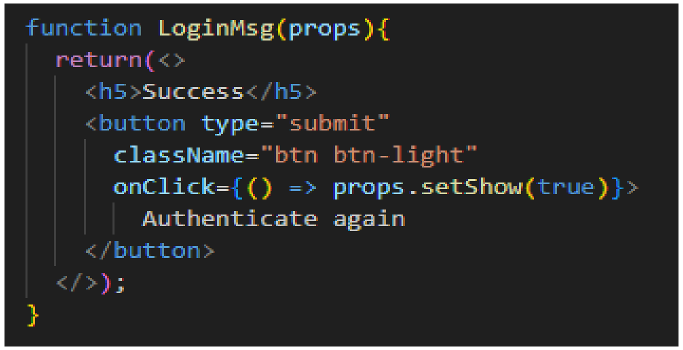


login.js

```
function Login(){
  const [show, setShow]     = React.useState(true);
  const [status, setStatus] = React.useState('');

  return (
    <Card
      bgcolor="secondary"
      header="Login"
      status={status}
      body={show ?
        <LoginForm setShow={setShow} setStatus={setStatus}/> :
        <LoginMsg setShow={setShow} setStatus={setStatus}/>}
    />
  )
}
```

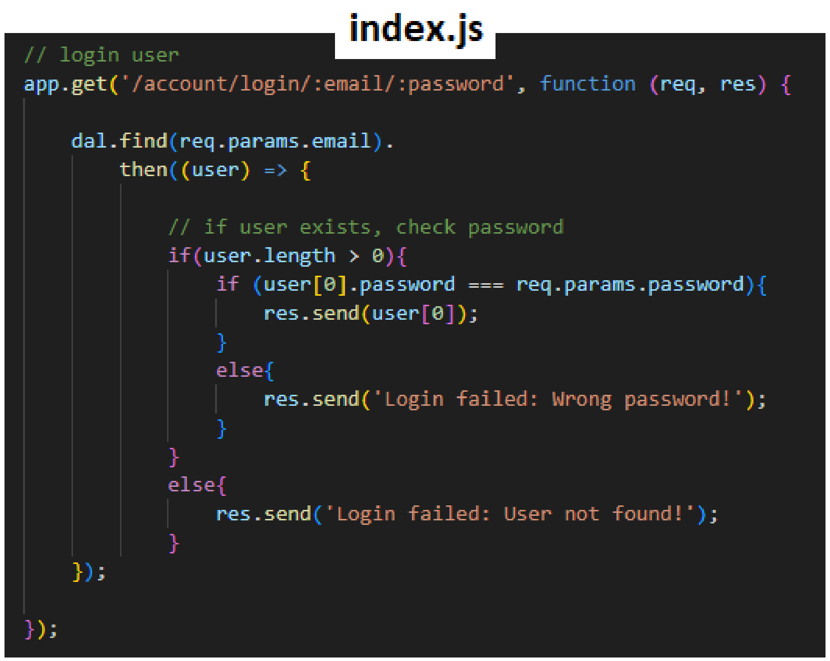


```
function handle(){
  console.log(email,password);
  if (!validate(email,    'email', props.setStatus))    return;
  if (!validate(password, 'password', props.setStatus)) return;

  fetch(`/account/login/${email}/${password}`)
  .then(response => response.text())
  .then(text => {
      try {
          const data = JSON.parse(text);
          props.setStatus('');
          props.setShow(false);
          console.log('JSON:', data);
      } catch(err) {
          props.setStatus(text)
          console.log('err:', text);
      }
  });
}
```



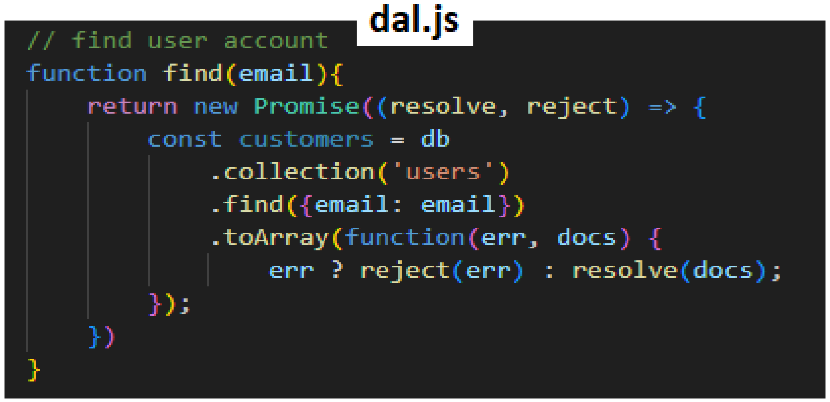
```
function LoginMsg(props){
  return(<>
    <h5>Success</h5>
    <button type="submit"
      className="btn btn-light"
      onClick={() => props.setShow(true)}>
        Authenticate again
    </button>
  </>);
}
```

{ REST-API }



index.js

```
// login user
app.get('/account/login/:email/:password', function (req, res) {

    dal.find(req.params.email).
        then((user) => {

            // if user exists, check password
            if(user.length > 0){
                if (user[0].password === req.params.password){
                    res.send(user[0]);
                }
                else{
                    res.send('Login failed: Wrong password!');
                }
            }
            else{
                res.send('Login failed: User not found!');
            }
    });

});
```

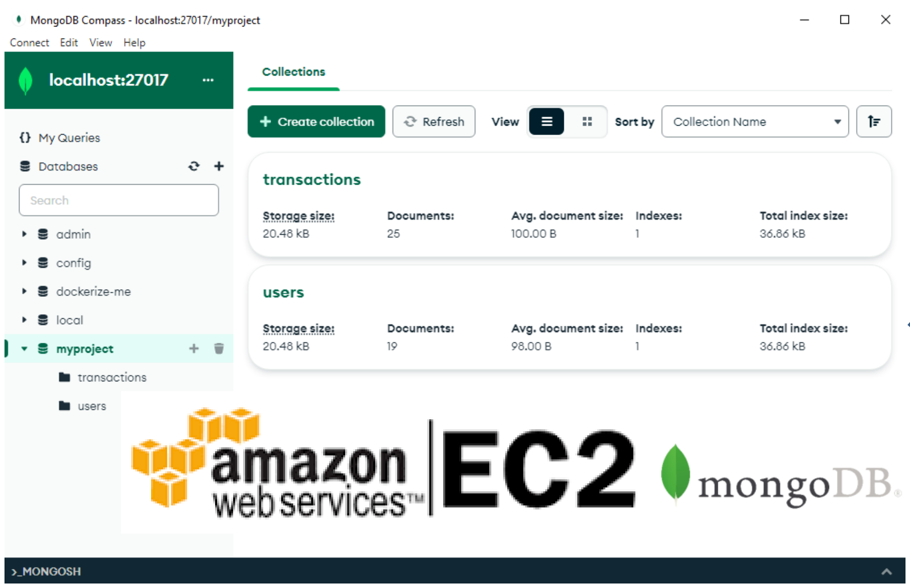


dal.js

```
// find user account
function find(email){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('users')
            .find({email: email})
            .toArray(function(err, docs) {
                err ? reject(err) : resolve(docs);
        });
    })
}
```

Part 2: Database And API

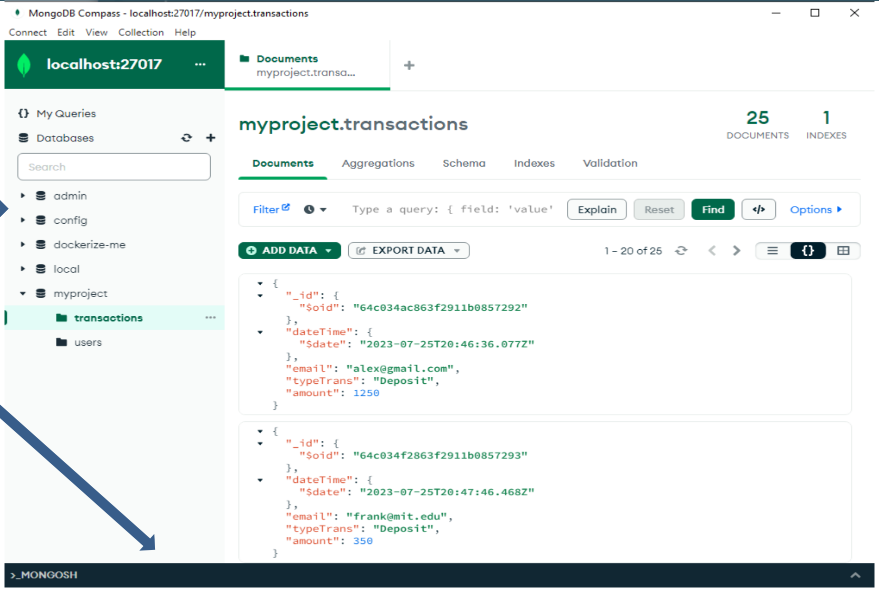
Database



MongoDB Compass - localhost:27017/myproject

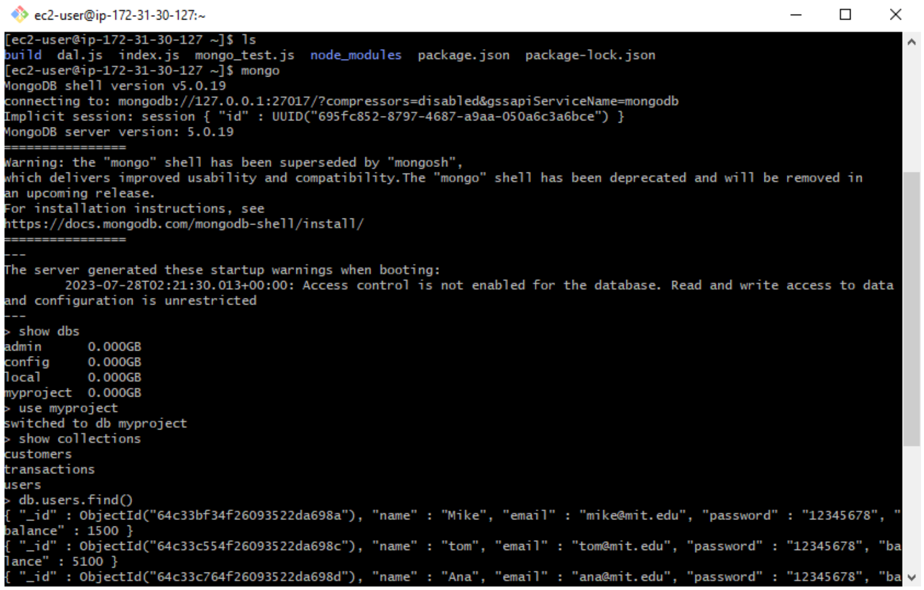
	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
transactions	20.48 kB	25	100.00 B	1	36.86 kB
users	20.48 kB	19	98.00 B	1	36.86 kB

amazon web services™ | EC2 mongoDB.



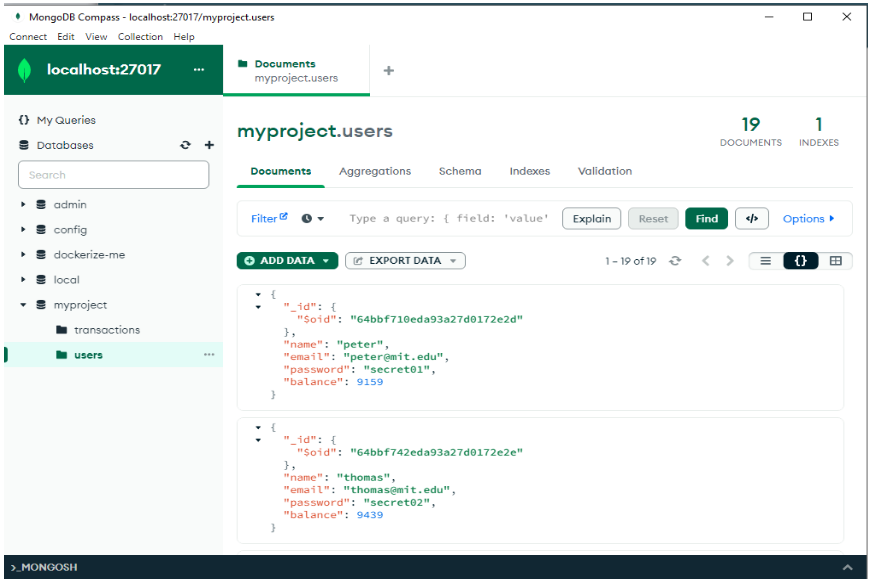
MongoDB Compass - localhost:27017/myproject.transactions

```
{
  "_id": {
    "$oid": "64c034ac863f2911b0857292"
  },
  "dateTime": {
    "$date": "2023-07-25T20:46:36.077Z"
  },
  "email": "alex@gmail.com",
  "typeTrans": "Deposit",
  "amount": 1250
}
{
  "_id": {
    "$oid": "64c034f2863f2911b0857293"
  },
  "dateTime": {
    "$date": "2023-07-25T20:47:46.468Z"
  },
  "email": "frank@mit.edu",
  "typeTrans": "Deposit",
  "amount": 350
}
```



ec2-user@ip-172-31-30-127:~

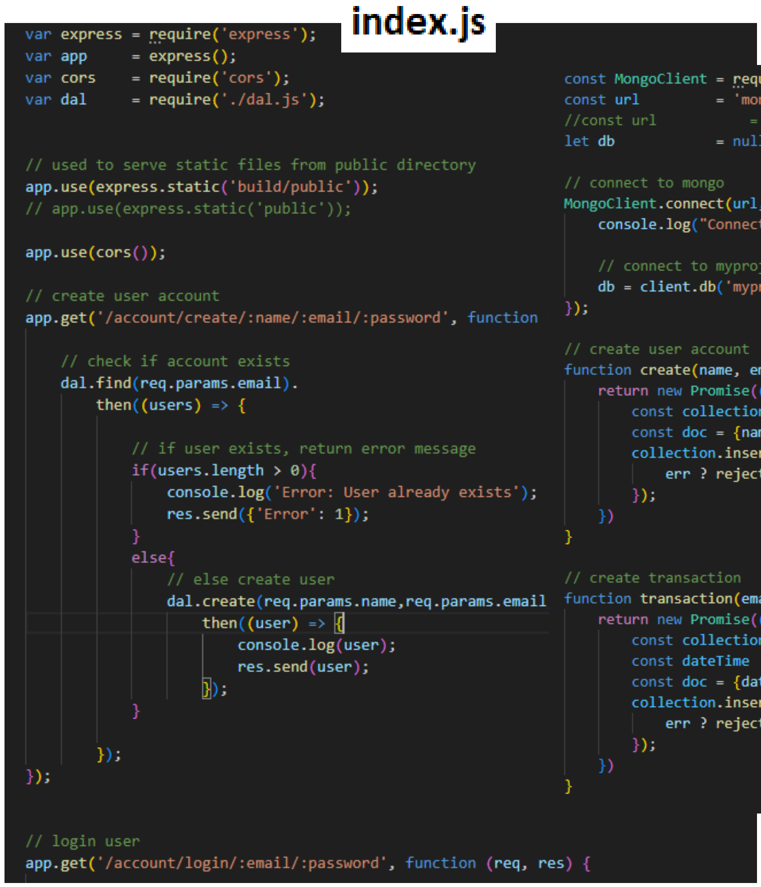
```
[ec2-user@ip-172-31-30-127 ~]$ ls
build  dal.js  index.js  mongo_test.js  node_modules  package.json  package-lock.json
[ec2-user@ip-172-31-30-127 ~]$ mongo
MongoDB shell version v5.0.19
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("695fc852-8797-4687-a9aa-050a6c3a6bce") }
MongoDB server version: 5.0.19
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2023-07-28T02:21:30.013+00:00: Access control is not enabled for the database. Read and write access to data
and configuration is unrestricted
---
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
myproject  0.000GB
> use myproject
switched to db myproject
> show collections
customers
transactions
users
> db.users.find()
{ "_id" : ObjectId("64c33bf34f26093522da698a"), "name" : "Mike", "email" : "mike@mit.edu", "password" : "12345678", "
balance" : 1500 }
{ "_id" : ObjectId("64c33c554f26093522da698c"), "name" : "tom", "email" : "tom@mit.edu", "password" : "12345678", "ba
lance" : 5100 }
{ "_id" : ObjectId("64c33c764f26093522da698d"), "name" : "Ana", "email" : "ana@mit.edu", "password" : "12345678", "ba
```



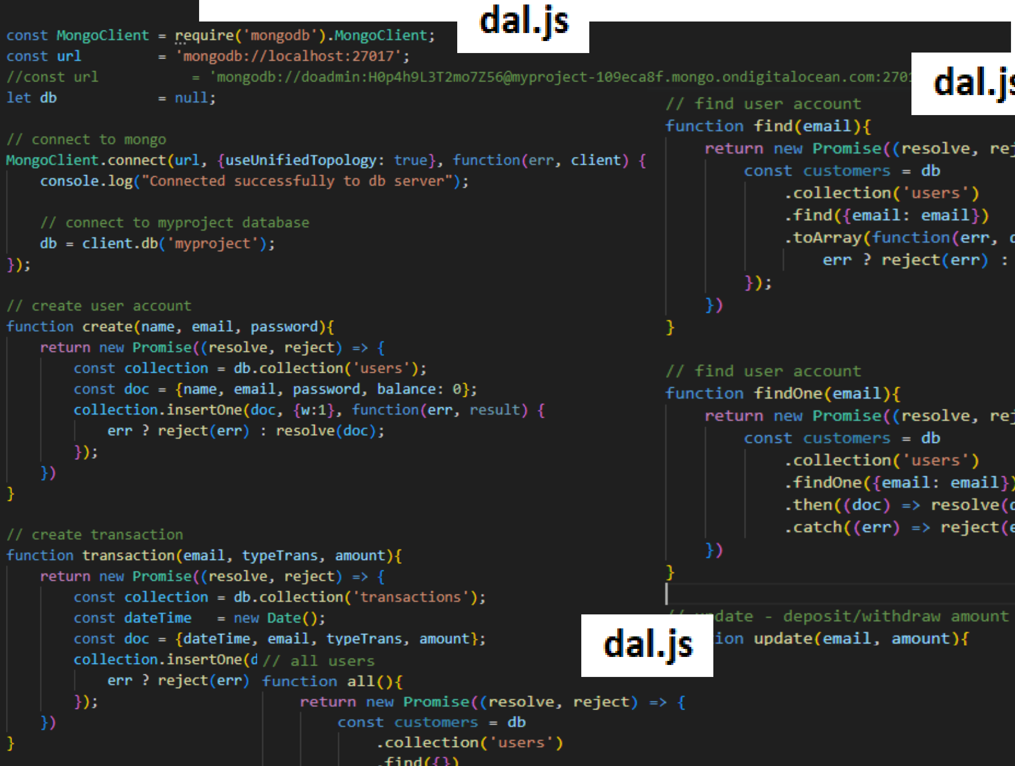
MongoDB Compass - localhost:27017/myproject.users

```
{
  "_id": {
    "$oid": "64bbf710eda93a27d0172e2d"
  },
  "name": "peter",
  "email": "peter@mit.edu",
  "password": "secret01",
  "balance": 9159
}
{
  "_id": {
    "$oid": "64bbf742eda93a27d0172e2e"
  },
  "name": "thomas",
  "email": "thomas@mit.edu",
  "password": "secret02",
  "balance": 9439
}
```

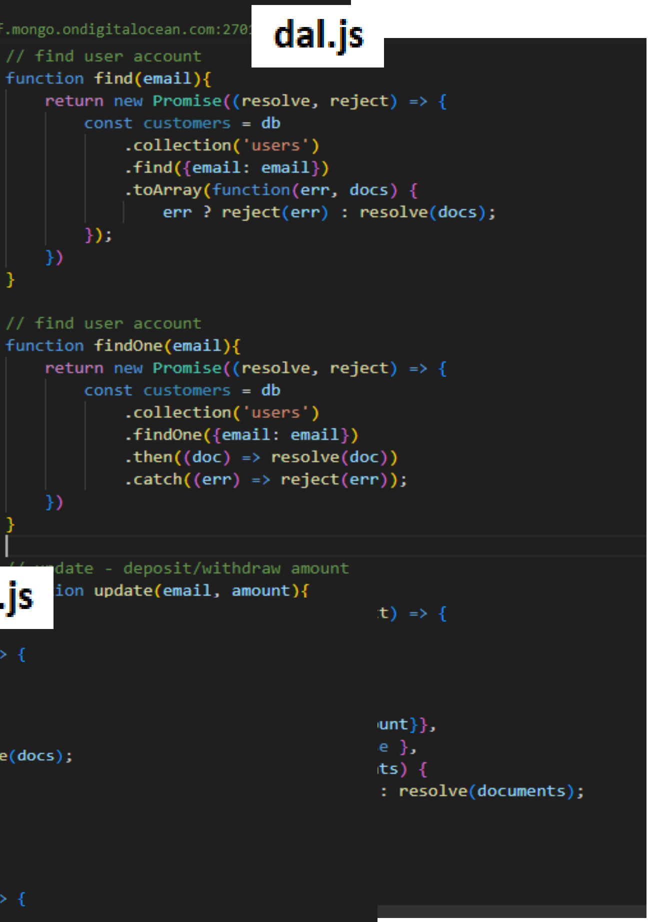
API


index.js

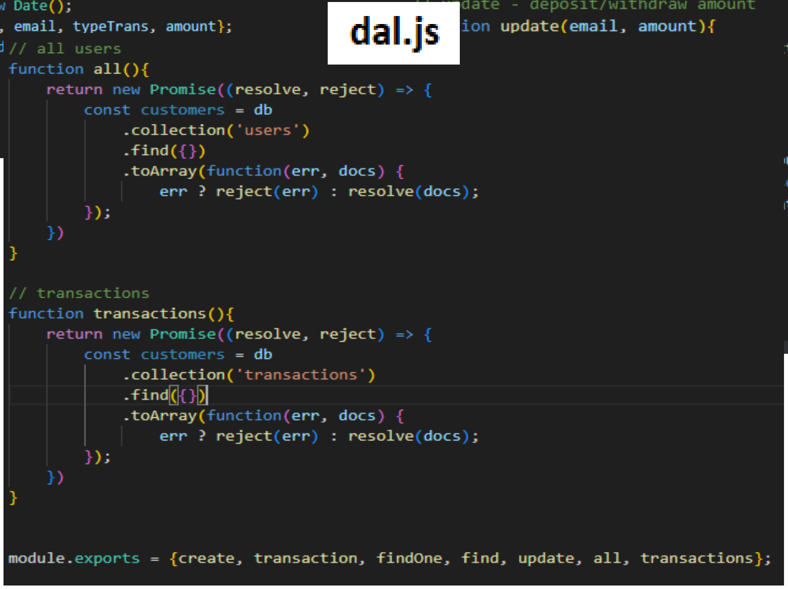
```
var express = require('express');
var app     = express();
var cors    = require('cors');
var dal     = require('./dal.js');


// used to serve static files from public directory
app.use(express.static('build/public'));
// app.use(express.static('public'));

app.use(cors());

// create user account
app.get('/account/create/:name/:email/:password', function

    // check if account exists
    dal.find(req.params.email).
        then((users) => {

            // if user exists, return error message
            if(users.length > 0){
                console.log('Error: User already exists');
                res.send({'Error': 1});
            }
            else{
                // else create user
                dal.create(req.params.name,req.params.email
                    then((user) => {
                        console.log(user);
                        res.send(user);
                    });
            }

        });
});


// login user
app.get('/account/login/:email/:password', function (req, res) {
```


dal.js

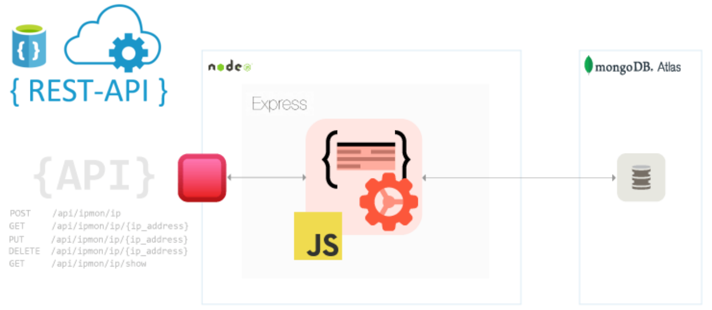
```
const MongoClient = require('mongodb').MongoClient;
const url         = 'mongodb://localhost:27017';
//const url            = 'mongodb://doadmin:H0p4h9L3T2mo7Z56@myproject-109eca8f.mongo.ondigitalocean.com:270
let db            = null;

// connect to mongo
MongoClient.connect(url, {useUnifiedTopology: true}, function(err, client) {
    console.log("Connected successfully to db server");

    // connect to myproject database
    db = client.db('myproject');
});

// create user account
function create(name, email, password){
    return new Promise((resolve, reject) => {
        const collection = db.collection('users');
        const doc = {name, email, password, balance: 0};
        collection.insertOne(doc, {w:1}, function(err, result) {
            err ? reject(err) : resolve(doc);
        });
    })
}

// create transaction
function transaction(email, typeTrans, amount){
    return new Promise((resolve, reject) => {
        const collection = db.collection('transactions');
        const dateTime   = new Date();
        const doc = {dateTime, email, typeTrans, amount};
        collection.insertOne(d
            err ? reject(err)
        });
    })
}
```


dal.js

```
// find user account
function find(email){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('users')
            .find({email: email})
            .toArray(function(err, docs) {
                err ? reject(err) : resolve(docs);
            });
    })
}

// find user account
function findOne(email){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('users')
            .findOne({email: email})
            .then((doc) => resolve(doc))
            .catch((err) => reject(err));
    })
}

  date - deposit/withdraw amount
 ion update(email, amount){
                                                t) => {

                                                unt}},
                                                e },
                                                ts) {
                                                : resolve(documents);
```


dal.js

```
// all users
function all(){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('users')
            .find({})
            .toArray(function(err, docs) {
                err ? reject(err) : resolve(docs);
        });
    })
}

// transactions
function transactions(){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('transactions')
            .find({})
            .toArray(function(err, docs) {
                err ? reject(err) : resolve(docs);
        });
    })
}


module.exports = {create, transaction, findOne, find, update, all, transactions};
```

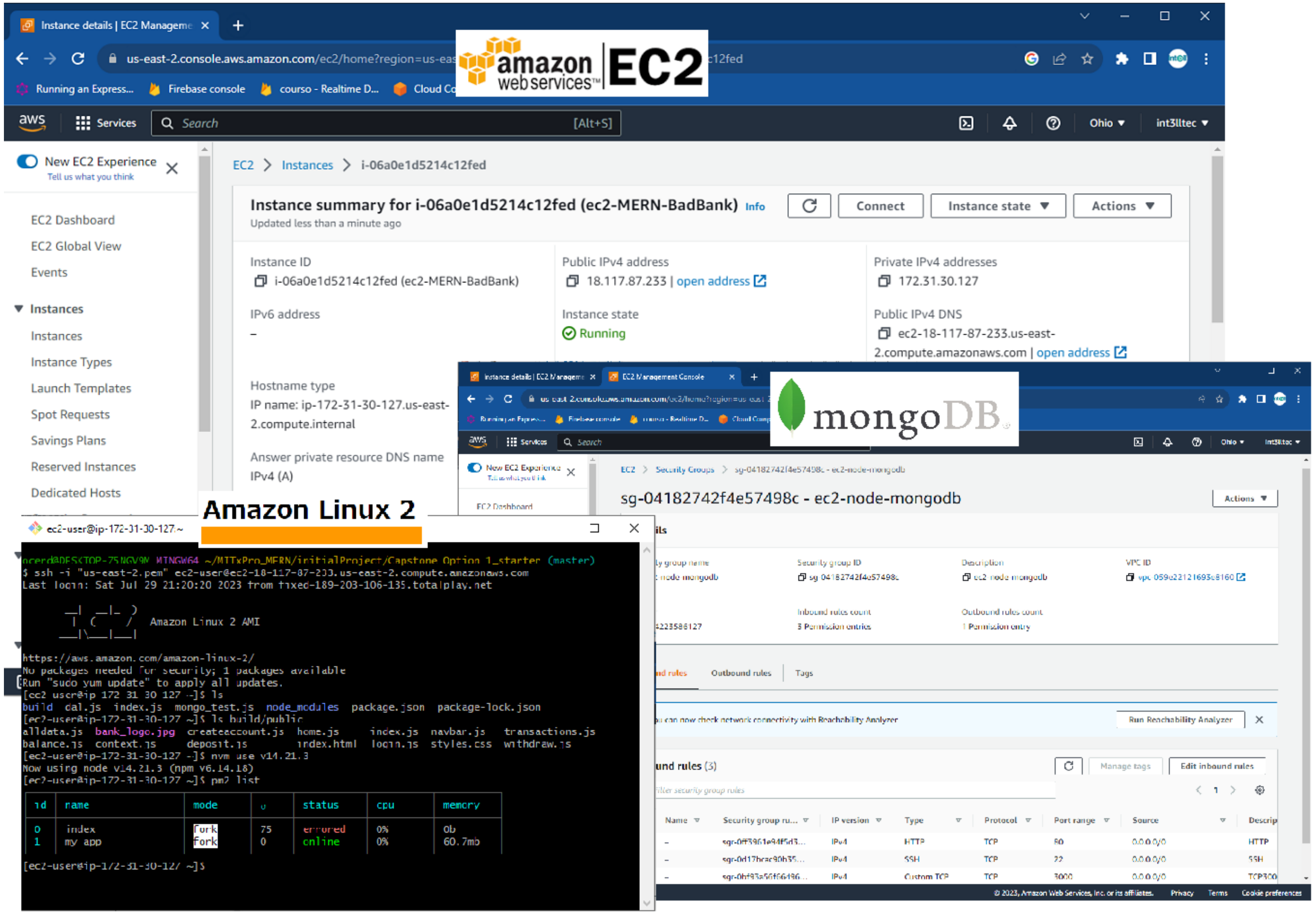


{ REST-API }
{API}
POST /api/ipmon/ip
GET /api/ipmon/ip/{ip_address}
PUT /api/ipmon/ip/{ip_address}
DELETE /api/ipmon/ip/{ip_address}
GET /api/ipmon/ip/show
node
Express
JS
mongoDB. Atlas

MIT xPRO

Part 3: Deployment, Additional Features, App Demonstration, Reflection

Deployment



Amazon Linux 2

Instance summary for i-06a0e1d5214c12fed (ec2-MERN-BadBank) Info
Updated less than a minute ago

Instance ID: i-06a0e1d5214c12fed (ec2-MERN-BadBank)
Public IPv4 address: 18.117.87.233 | open address
Private IPv4 addresses: 172.31.30.127
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-18-117-87-233.us-east-2.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-30-127.us-east-2.compute.internal
Answer private resource DNS name: IPv4 (A)

```
ocerd@DESKTOP-75NGV9M MINGW64 ~/MITxPro_MERN/initialProject/Capstone Option 1_starter (master)
$ ssh -i "us-east-2.pem" ec2-user@ec2-18-117-87-233.us-east-2.compute.amazonaws.com
Last login: Sat Jul 29 21:20:20 2023 from fixed-189-203-106-135.totalplay.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-127 ~]$ ls
build  dal.js  index.js  mongo_test.js  node_modules  package.json  package-lock.json
[ec2-user@ip-172-31-30-127 ~]$ ls build/public
alldata.js  bank_logo.jpg  createaccount.js  home.js     index.js  navbar.js   transactions.js
balance.js  context.js     deposit.js        index.html  login.js  styles.css  withdraw.js
[ec2-user@ip-172-31-30-127 ~]$ nvm use v14.21.3
Now using node v14.21.3 (npm v6.14.18)
[ec2-user@ip-172-31-30-127 ~]$ pm2 list
│ id │ name     │ mode │ ↺  │ status  │ cpu │ memory │
│ 0  │ index    │ fork │ 75 │ errored │ 0%  │ 0b     │
│ 1  │ my app   │ fork │ 0  │ online  │ 0%  │ 60.7mb │
[ec2-user@ip-172-31-30-127 ~]$
```

sg-04182742f4e57498c - ec2-node-mongodb

Security group ID: sg-04182742f4e57498c
Description: ec2-node-mongodb
VPC ID: vpc-059e22121693e8160
Inbound rules count: 3 Permission entries
Outbound rules count: 1 Permission entry

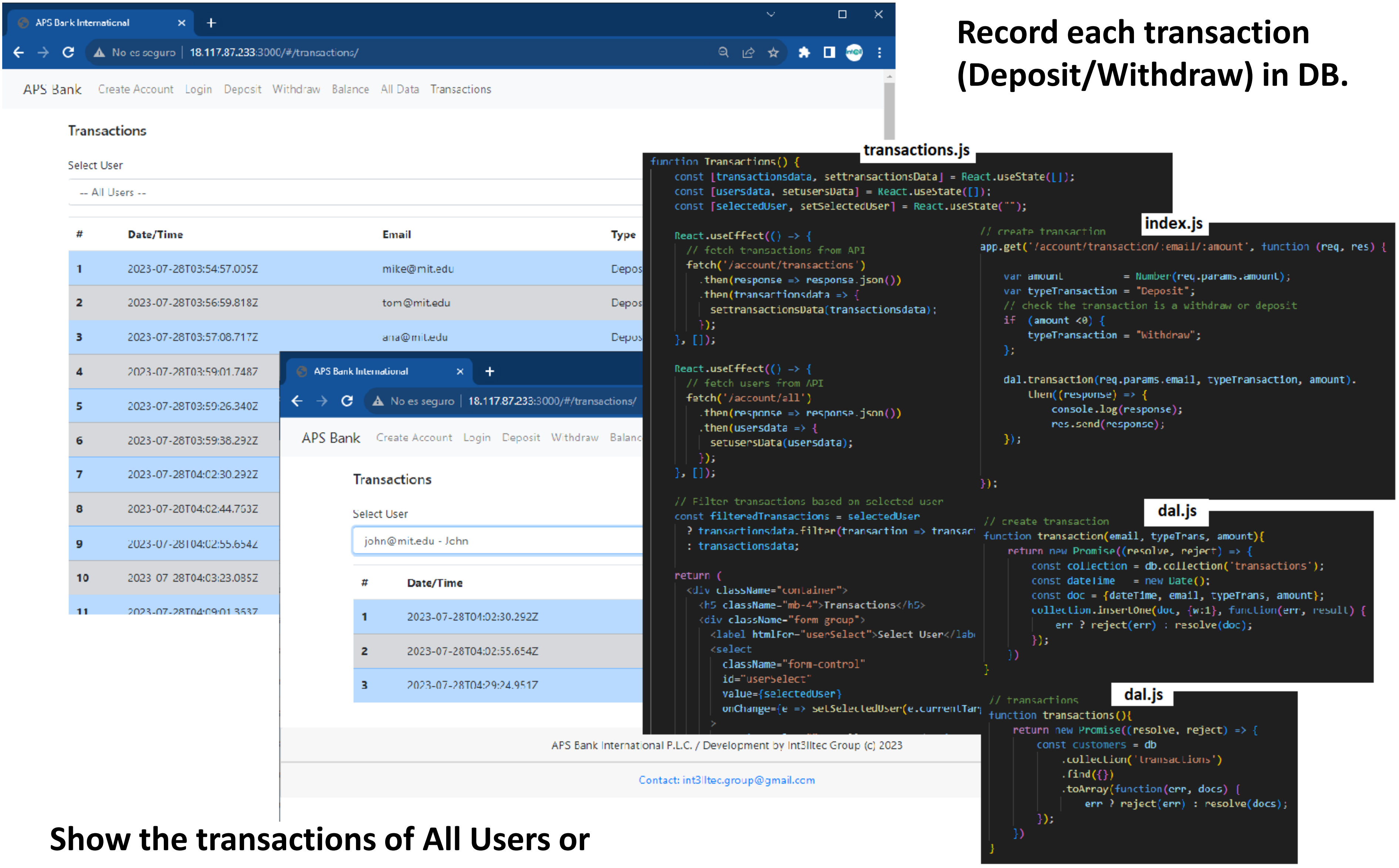
Inbound rules (3)

Name	Security group ru...	IP version	Type	Protocol	Port range	Source	Descrip
–	sgr-0ff3961e94f5d3...	IPv4	HTTP	TCP	80	0.0.0.0/0	HTTP
–	sgr-0d17bcac90b35...	IPv4	SSH	TCP	22	0.0.0.0/0	SSH
–	sgr-0bf93a56f66496...	IPv4	Custom TCP	TCP	3000	0.0.0.0/0	TCP300

© 2023, Amazon Web Services, Inc. or its affiliates.

MIT xPRO

Additional Features



APS Bank International
No es seguro | 18.117.87.233:3000/#/transactions/
APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions

Transactions

Select User

-- All Users --

#	Date/Time	Email	Type
1	2023-07-28T03:54:57.005Z	mike@mit.edu	Depos
2	2023-07-28T03:56:59.818Z	tom@mit.edu	Depos
3	2023-07-28T03:57:08.717Z	ana@mit.edu	Depos
4	2023-07-28T03:59:01.748Z		
5	2023-07-28T03:59:26.340Z		
6	2023-07-28T03:59:38.292Z		
7	2023-07-28T04:02:30.292Z		
8	2023-07-28T04:02:44.753Z		
9	2023-07-28T04:02:55.654Z		
10	2023-07-28T04:03:23.085Z		
11	2023-07-28T04:09:01.353Z		

APS Bank International
No es seguro | 18.117.87.233:3000/#/transactions/
APS Bank Create Account Login Deposit Withdraw Balanc

Transactions

Select User

john@mit.edu - John

#	Date/Time
1	2023-07-28T04:02:30.292Z
2	2023-07-28T04:02:55.654Z
3	2023-07-28T04:29:24.951Z

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com

transactions.js

```
function Transactions() {
    const [transactionsdata, settransactionsData] = React.useState([]);
    const [usersdata, setusersData] = React.useState([]);
    const [selectedUser, setSelectedUser] = React.useState("");

    React.useEffect(() => {
      // fetch transactions from API
      fetch('/account/transactions')
        .then(response => response.json())
        .then(transactionsdata => {
          settransactionsData(transactionsdata);
        });
    }, []);

    React.useEffect(() => {
      // fetch users from API
      fetch('/account/all')
        .then(response => response.json())
        .then(usersdata => {
          setusersData(usersdata);
        });
    }, []);

    // Filter transactions based on selected user
    const filteredTransactions = selectedUser
      ? transactionsdata.filter(transaction => transac
      : transactionsdata;

    return (
      <div className="container">
        <h5 className="mb-4">Transactions</h5>
        <div className="form group">
          <label htmlFor="userSelect">Select User</labe
          <select
            className="form-control"
            id="userSelect"
            value={selectedUser}
            onChange={e => setSelectedUser(e.currentTar
          >
```

index.js

```
// create transaction
app.get('/account/transaction/:email/:amount', function (req, res) {

    var amount          = Number(req.params.amount);
    var typeTransaction = "Deposit";
    // check the transaction is a withdraw or deposit
    if  (amount <0) {
        typeTransaction = "Withdraw";
    };

    dal.transaction(req.params.email, typeTransaction, amount).
        then((response) => {
            console.log(response);
            res.send(response);
    });


});
```

dal.js

```
// create transaction
function transaction(email, typeTrans, amount){
    return new Promise((resolve, reject) => {
        const collection = db.collection('transactions');
        const dateTime   = new Date();
        const doc = {dateTime, email, typeTrans, amount};
        collection.insertOne(doc, {w:1}, function(err, result) {
            err ? reject(err) : resolve(doc);
        });
    })
}
```

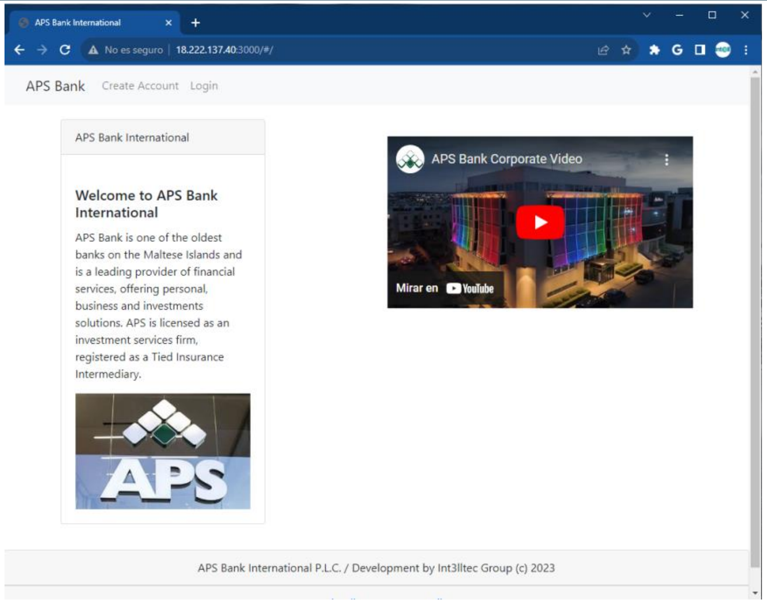
dal.js

```
// transactions
function transactions(){
    return new Promise((resolve, reject) => {
        const customers = db
            .collection('transactions')
            .find({})
            .toArray(function(err, docs) {
                err ? reject(err) : resolve(docs);
        });
    })
}
```

Record each transaction (Deposit/Withdraw) in DB.

Show the transactions of All Users or Selected user

Application Demonstration – Create Account, Log In, Deposit, Withdraw



APS Bank International

No es seguro | 18.222.137.40:3000/#/

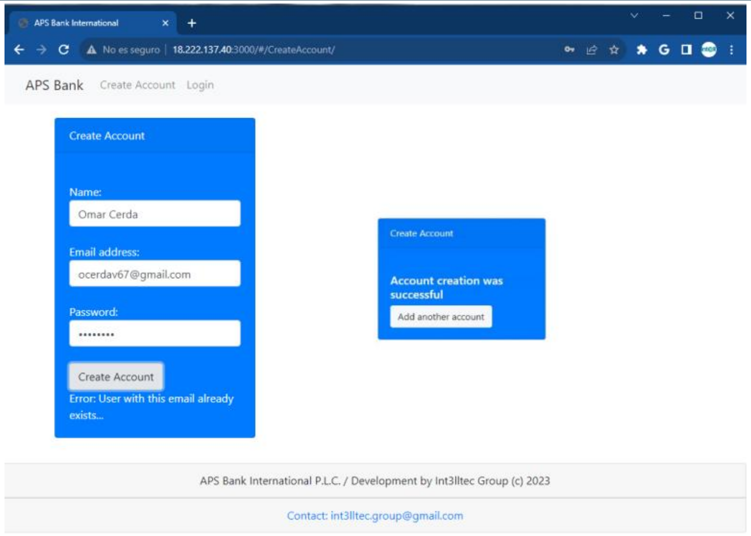
APS Bank Create Account Login

APS Bank International

Welcome to APS Bank International

APS Bank is one of the oldest banks on the Maltese Islands and is a leading provider of financial services, offering personal, business and investments solutions. APS is licensed as an investment services firm, registered as a Tied Insurance Intermediary.

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023



APS Bank International

No es seguro | 18.222.137.40:3000/#/CreateAccount/

APS Bank Create Account Login

Create Account

Name: Omar Cerda

Email address: ocerdav67@gmail.com

Password: ••••••••

Create Account

Error: User with this email already exists...

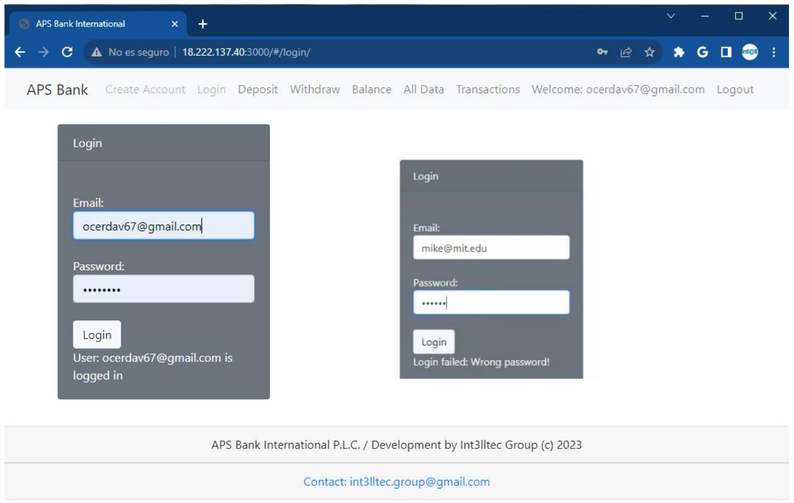
Create Account

Account creation was successful

Add another account

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com



APS Bank International

No es seguro | 18.222.137.40:3000/#/login/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

Login

Email: ocerdav67@gmail.com

Password: ••••••••

Login

User: ocerdav67@gmail.com is logged in

Login

Email: mike@mit.edu

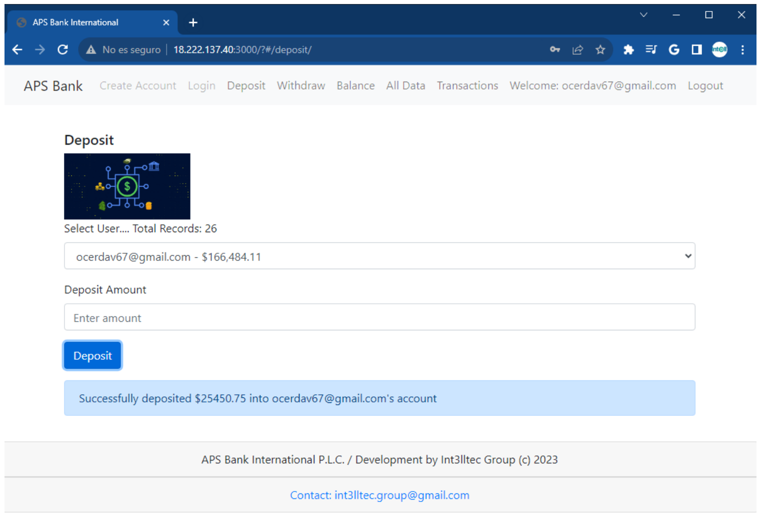
Password: ••••••

Login

Login failed: Wrong password!

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com



APS Bank International

No es seguro | 18.222.137.40:3000/?#/deposit/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

Deposit

Select User.... Total Records: 26

ocerdav67@gmail.com - $166,484.11

Deposit Amount

Enter amount

Deposit

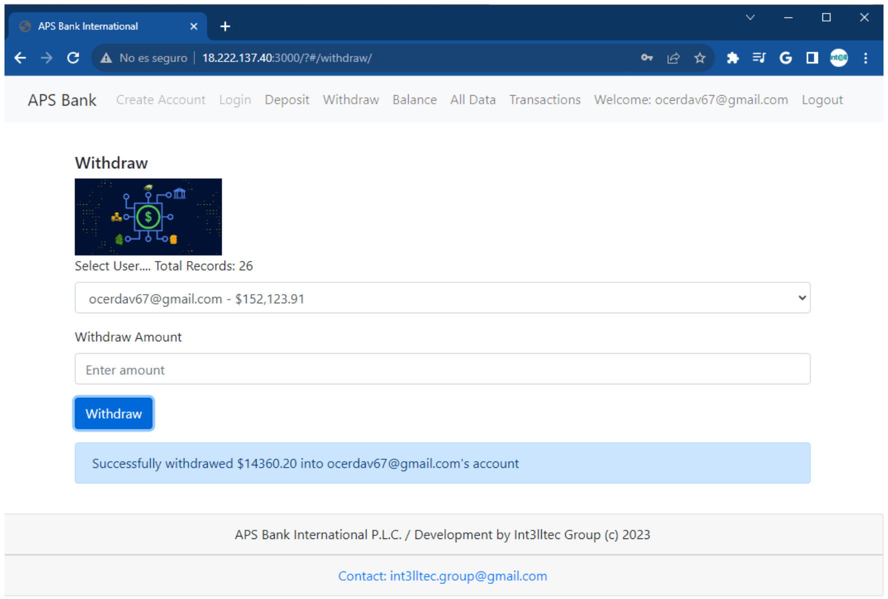
Successfully deposited $25450.75 into ocerdav67@gmail.com's account

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com

MIT xPRO

Application Demonstration – Create Account, Log In, Deposit, Withdraw



APS Bank International

No es seguro | 18.222.137.40:3000/?#/withdraw/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

Withdraw

Select User.... Total Records: 26

ocerdav67@gmail.com - $152,123.91

Withdraw Amount

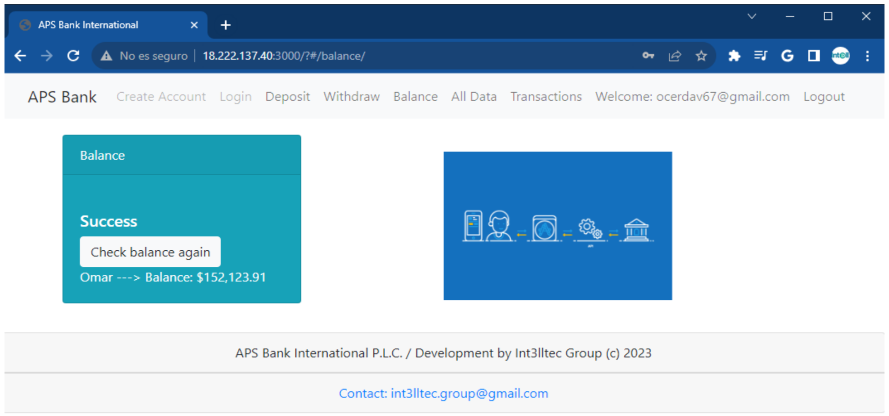
Enter amount

Withdraw

Successfully withdrawed $14360.20 into ocerdav67@gmail.com's account

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com



APS Bank International

No es seguro | 18.222.137.40:3000/?#/balance/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

Balance

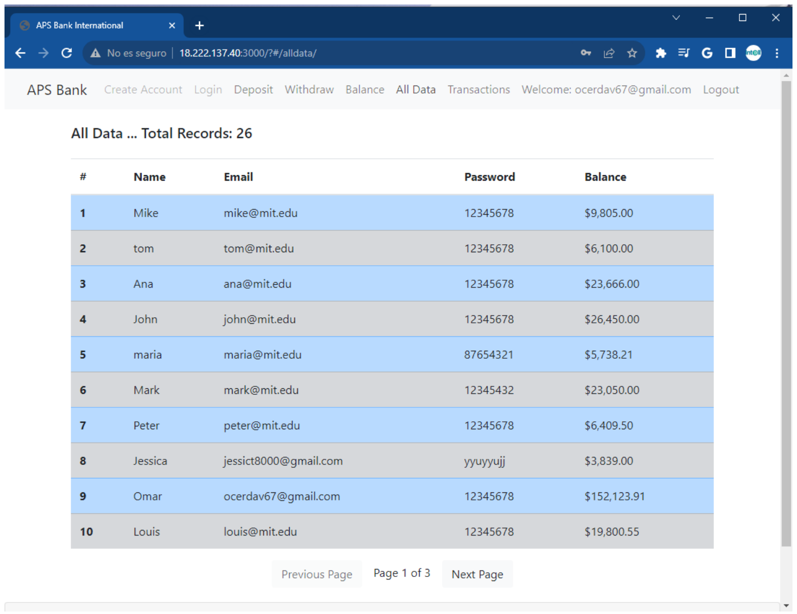
Success

Check balance again

Omar ---> Balance: $152,123.91

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com



APS Bank International

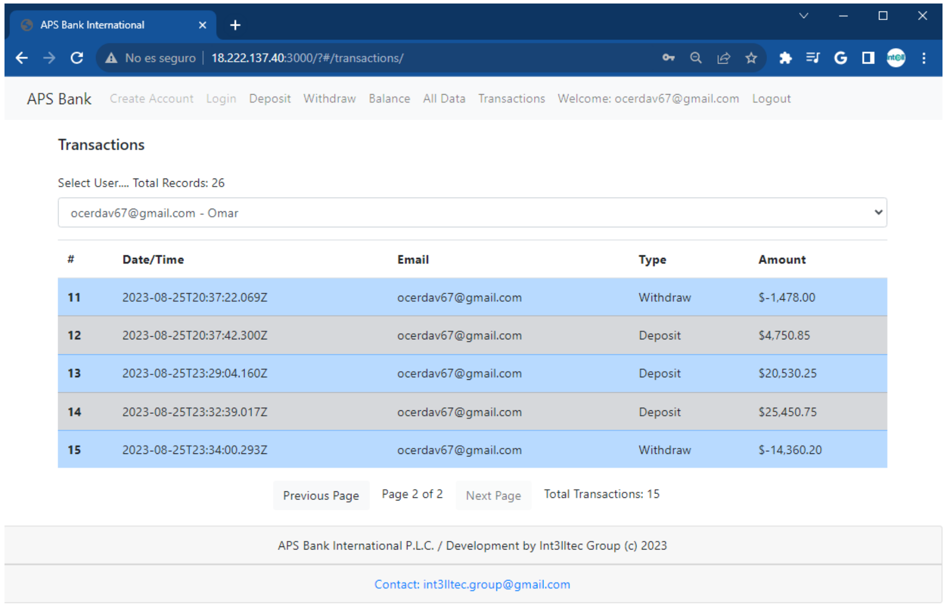
No es seguro | 18.222.137.40:3000/?#/alldata/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

All Data ... Total Records: 26

#	Name	Email	Password	Balance
1	Mike	mike@mit.edu	12345678	$9,805.00
2	tom	tom@mit.edu	12345678	$6,100.00
3	Ana	ana@mit.edu	12345678	$23,666.00
4	John	john@mit.edu	12345678	$26,450.00
5	maria	maria@mit.edu	87654321	$5,738.21
6	Mark	mark@mit.edu	12345432	$23,050.00
7	Peter	peter@mit.edu	12345678	$6,409.50
8	Jessica	jessict8000@gmail.com	yyuyyujj	$3,839.00
9	Omar	ocerdav67@gmail.com	12345678	$152,123.91
10	Louis	louis@mit.edu	12345678	$19,800.55

Previous Page Page 1 of 3 Next Page



APS Bank International

No es seguro | 18.222.137.40:3000/?#/transactions/

APS Bank Create Account Login Deposit Withdraw Balance All Data Transactions Welcome: ocerdav67@gmail.com Logout

Transactions

Select User.... Total Records: 26

ocerdav67@gmail.com - Omar

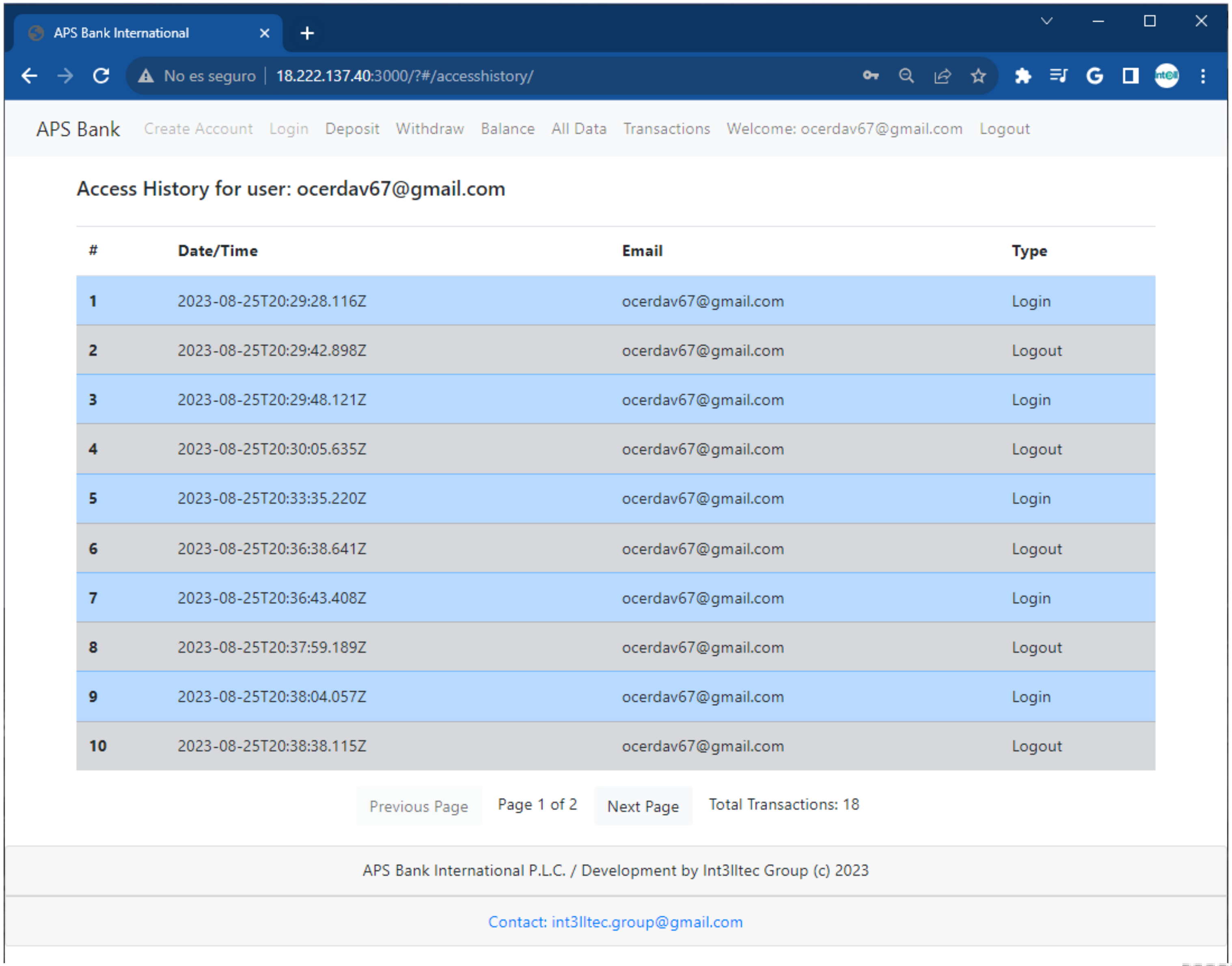
#	Date/Time	Email	Type	Amount
11	2023-08-25T20:37:22.069Z	ocerdav67@gmail.com	Withdraw	$-1,478.00
12	2023-08-25T20:37:42.300Z	ocerdav67@gmail.com	Deposit	$4,750.85
13	2023-08-25T23:29:04.160Z	ocerdav67@gmail.com	Deposit	$20,530.25
14	2023-08-25T23:32:39.017Z	ocerdav67@gmail.com	Deposit	$25,450.75
15	2023-08-25T23:34:00.293Z	ocerdav67@gmail.com	Withdraw	$-14,360.20

Previous Page Page 2 of 2 Next Page Total Transactions: 15

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com

Application Demonstration – Create Account, Log In, Deposit, Withdraw



Access History for user: ocerdav67@gmail.com

#	Date/Time	Email	Type
1	2023-08-25T20:29:28.116Z	ocerdav67@gmail.com	Login
2	2023-08-25T20:29:42.898Z	ocerdav67@gmail.com	Logout
3	2023-08-25T20:29:48.121Z	ocerdav67@gmail.com	Login
4	2023-08-25T20:30:05.635Z	ocerdav67@gmail.com	Logout
5	2023-08-25T20:33:35.220Z	ocerdav67@gmail.com	Login
6	2023-08-25T20:36:38.641Z	ocerdav67@gmail.com	Logout
7	2023-08-25T20:36:43.408Z	ocerdav67@gmail.com	Login
8	2023-08-25T20:37:59.189Z	ocerdav67@gmail.com	Logout
9	2023-08-25T20:38:04.057Z	ocerdav67@gmail.com	Login
10	2023-08-25T20:38:38.115Z	ocerdav67@gmail.com	Logout

Total Transactions: 18

APS Bank International P.L.C. / Development by Int3lltec Group (c) 2023

Contact: int3lltec.group@gmail.com

Reflection

- If you started the project today, how would you structure your code differently?

If I started the project today, I would structure the code differently by following a more modular and scalable architecture. I would use design patterns such as **MVC (Model-View-Controller)** to clearly separate **business logic**, **views**, and **interactions with the database**. Additionally, I would implement the use of **controllers to handle different server routes and actions**. I would also use **TypeScript** to add static typing and improve code maintainability and robustness. Moreover, I would implement **unit and integration tests** to ensure more reliable code.

- If you started the project today, what additional features would you build?

If I started the project today, I would add features such as **two-factor authentication (2FA)** to enhance the security of user accounts. I would also implement real-time notifications so that users receive instant alerts about their transactions. **Another interesting feature would be to allow users to set savings goals and receive notifications when they are close to achieving them**. Additionally, I would consider adding support for **transfers between accounts** and the ability to link external bank accounts for comprehensive financial management.

MIT | xPRO