



REFACTOR

BANKING APPLICATION

By Kate Post and Aly Westwood

Jan 2022



HIGH LEVEL BACKGROUND

REFACTOR

ONLINE BANKING

- Create accounts
- View funds
- Make Transfers
- Deposit Cheques

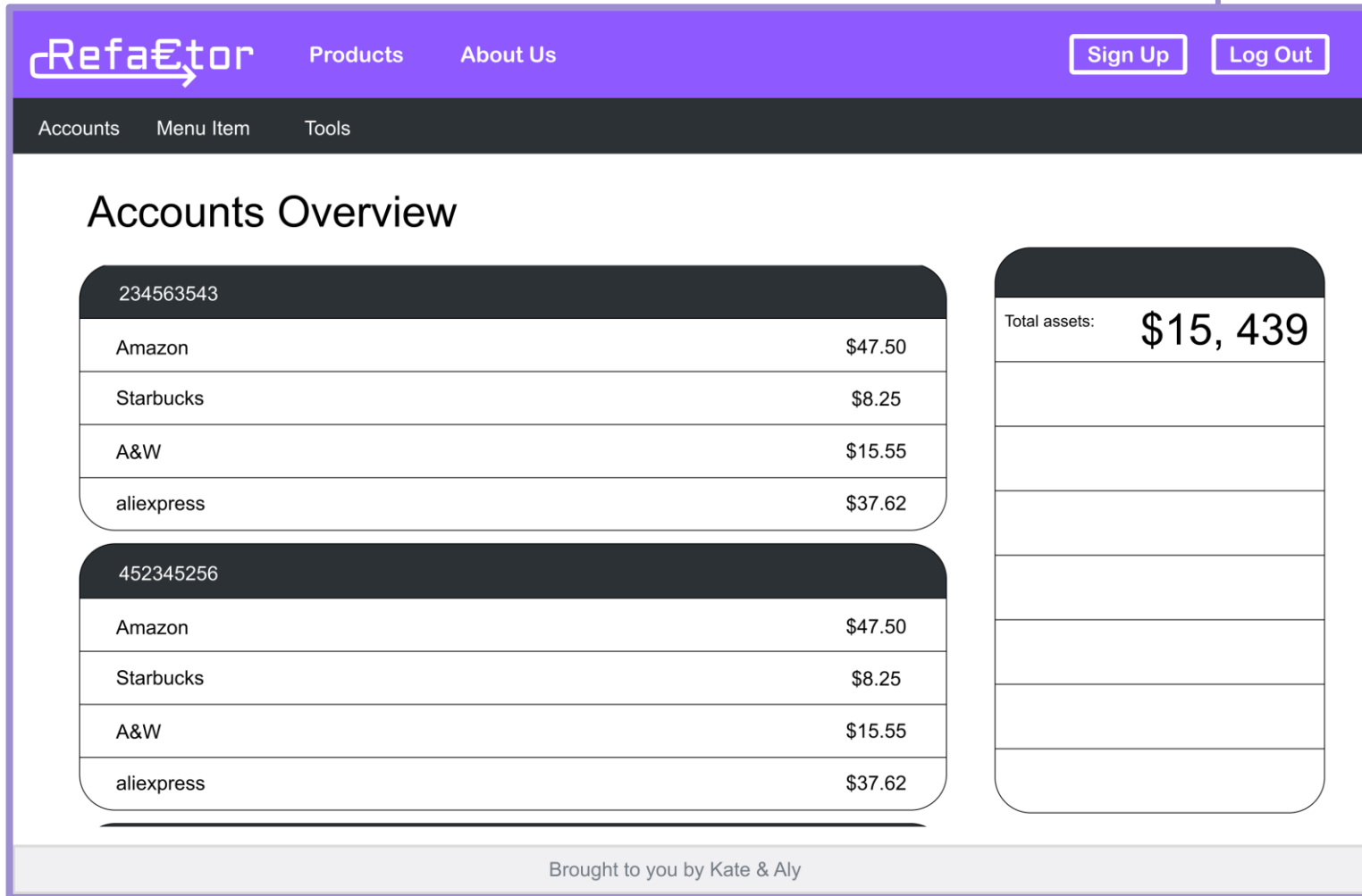
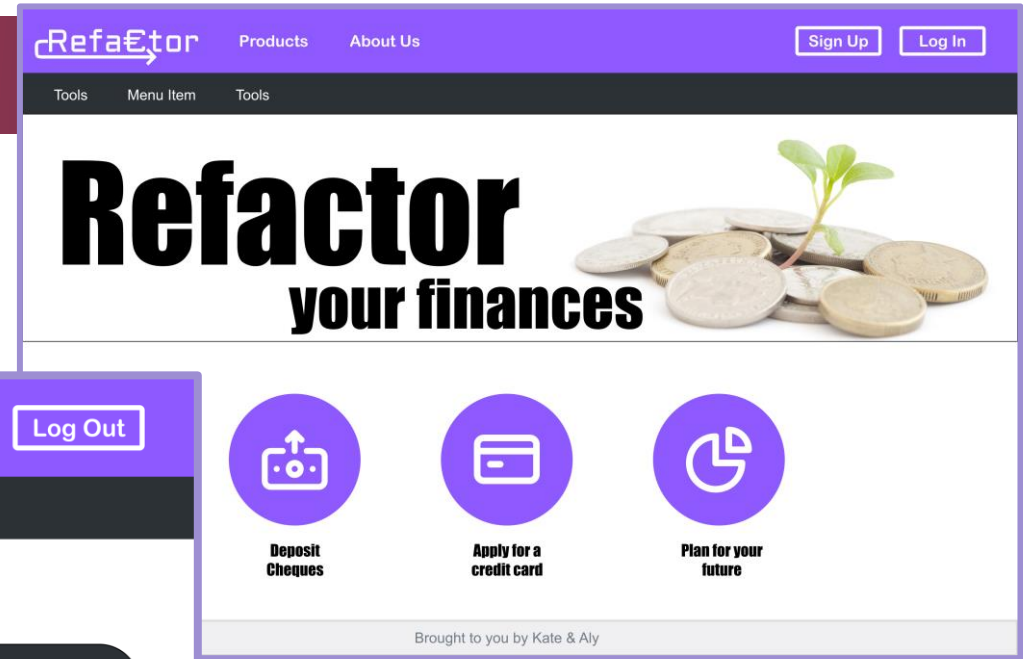




OUR SOLUTION

REFACTOR

DESIGN



USERS

Totals

Grand total assets
CAD \$1532.59

Total CAD \$1388.27

Total USD \$0.00

Total EUR €100.96

Total GBP £0.00

Your Information

Name: Kate Post

Date of Birth: 1991-12-12

SIN: *****3987

Email: kate@k.com

Phone Number: 5147920058

Update Your Contact Information

Update Email Address

kate@k.com

Update Phone Number

5147920058

Update Contact Info

Update Your Password

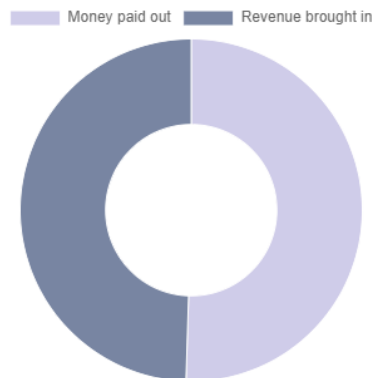
Current Password

New Password

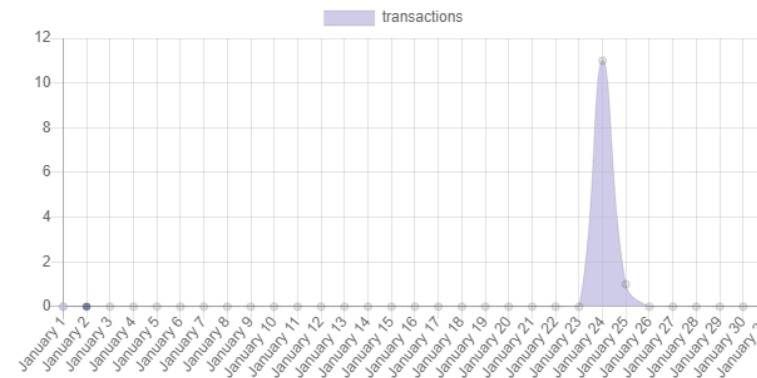
New Password Repeat

Update Password

Your data at a glance



Total money withdrawn and deposited in January



Daily Transactions in January

TRANSACTION BETWEEN USERS

Deposit a cheque

Account

Select

Account number on Cheque

Value

Cheque S/N

Upload image

Choose File

No file chosen

Send

Transaction # 4

Payer Account #	34
Payee Account #	4
Time Stamp	23 January 2022, 7:00 pm
Payer Account Value	\$325.50 CAD
Payee Account Value	\$325.50 CAD
Status	accepted
Recurring Payment	No

24 January 2022, 4:23 pm : paid to #14	(\$20.00 CAD)
24 January 2022, 4:24 pm : received from #14	\$17.53 CAD
24 January 2022, 5:30 pm : received from #44	pending \$56.31 CAD
24 January 2022, 7:00 pm : received from #44	pending \$36.02 CAD

Previous 1 2 Next

CREDIT ACCOUNTS

Credit Account Information

Credit Limit:
\$1500.00

Next Payment Due Date:
2022-02-07

Minimum Payment Due Now:
\$0.00

Interest Rate:
15.3000% APR

Late Payment Fees Due:
\$0.00

[Pay Late Payment Fees](#)

Credit - CAD

Account #: 34

Balance: \$402.64

24 January 2022, 4:17 pm : paid to #4	(\$325.50 CAD)
24 January 2022, 4:21 pm : paid to #14	(\$100.00 CAD)
24 January 2022, 4:21 pm : paid to #4	(\$2.50 CAD)
24 January 2022, 4:23 pm : received from #4	\$25.36 CAD

[4 total transactions](#)

Credit - CAD

Account #: 74

Pending Approval

Pay Late Payment Fees

From Account

Account #4 - CAD debit

Account with fees to pay

Account #34 - CAD credit

Late Payment Fees

\$16.00

Payment

\$ 5.50

[Make Transfer](#)

ADMINS CONTROL APPROVAL

Cat, Cassie

Status inactive
Role client
Email cassie@cat.com
Phone 5478962541
Birthdate 1998-03-12
Registration date 24 January 2022, 9:36 pm

Activate User

Disable

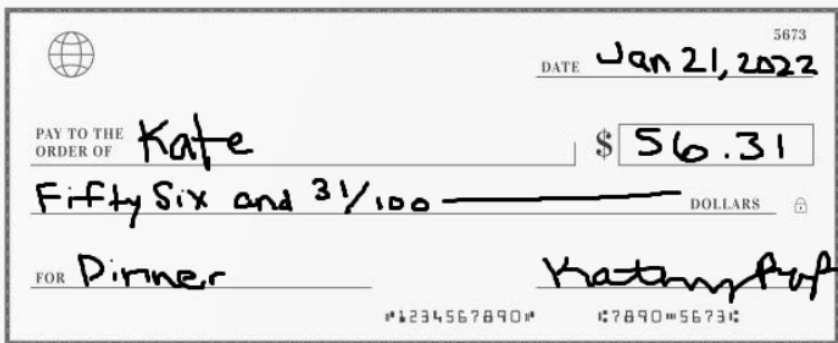
Account # 74 - application by client # 24

Client's total Assets: \$1532.58 CAD

Accounts Dossier

Account	Balance
CAD - debit	\$255.17
EUR - debit	€125.96
CAD - credit	\$1097.36
CAD - credit	\$0

Transaction # 4



Payee name: Kate Post

Upload date: 2022-01-24

Value on cheque: 56.31

Payer account number 44

Cheque number 5673

Approve Cheque

Deny Cheque

Credit Limit \$ CAD

1000

Interest Rate %

19.59

Approve

Reject

RECURRING PAYMENTS

- Checked daily
- From any account to any other account

Set up a recurring payment

From Account

Account #4 - CAD debit

Pay To

Account Number

Value

\$ Amount to be withdrawn

Repeating Interval

Number of days between payments

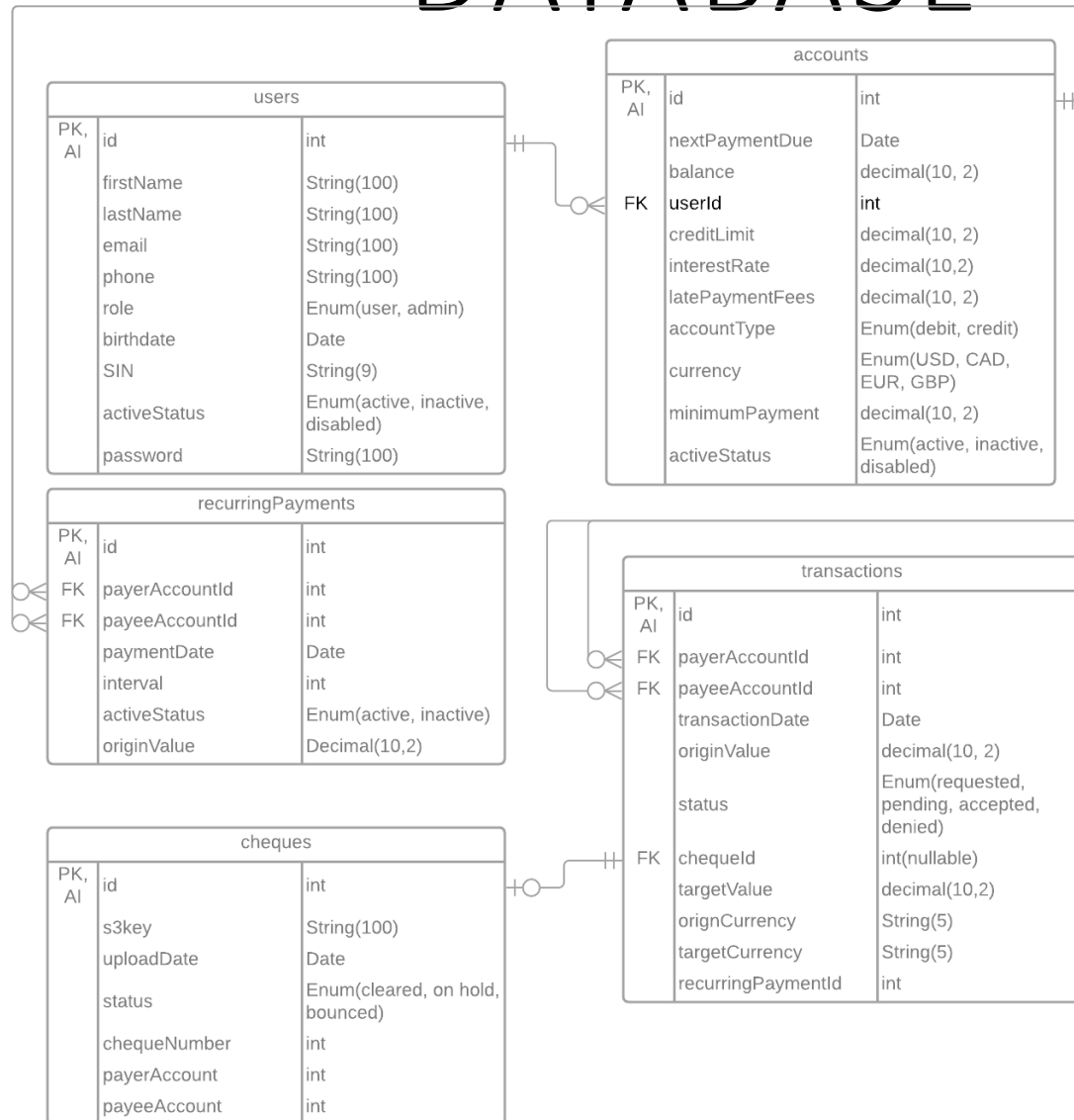
Payment start date

dd-----yyyy

Create recurring payment

From Account	To Account #	Amount	Interval	Next Payment	
4 - \$CAD debit	34	\$25.00	biweekly	26 January 2022	Cancel

DATABASE





CHALLENGES AND SOLUTIONS

ASYNCHRONOUS-NESS

```
function Client() {
  const state = useContext(AuthContext);
  if(state.authState === false){
    return <Navigate to='/login'/>
  }
  if(state.authState === true){
    if(state.authUser.role === 'admin'){
      return <Navigate to='/admin'/>
    }
    if(state.authUser.role === 'client'){
      switch(state.authUser.status){
        case 'disabled':
          return <Navigate to='/login'/>
        case 'inactive':
          return <div>Sit tight!<br/>You haven't been approved yet</div>
        case 'active':
          return <Outlet />
        default:
          break;
      }
    }
  }
  return <div>Loading...</div>
}
```

```
</Card.Header>
<ListGroup>
  {userList && userList.map((user) => {
    return (
      <ListGroup.Item
        key={"user" + user.id}
        onClick={() => setShowModal(user.id)}
      >
        <Row>
          <Col>{user.lastName}</Col>
```

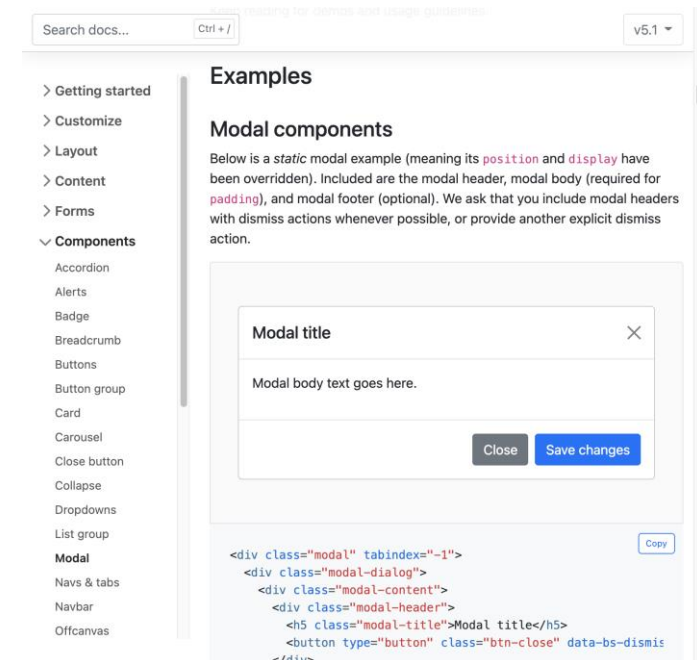
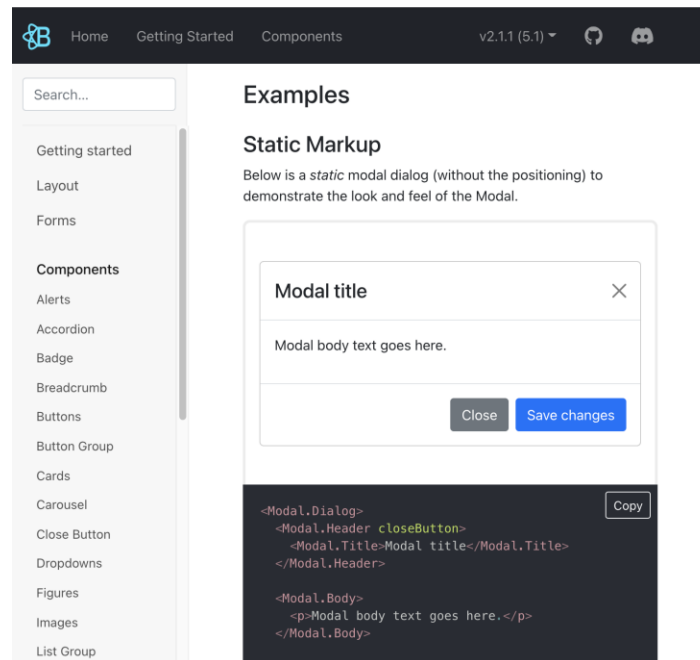
Consideration
must be made for
unset states

MODALS & REACT-BOOTSTRAP

```
<Modal
  key={"userModal" + user.id}
  size="lg"
  show={showModal === user.id}
  onHide={() => setShowModal(false)}
  aria-labelledby={"user" + user.id}
  centered
>
<Modal.Header
  variant="white"
  closeButton
  closeVariant="white"
  className="bg-secondary text-light"
>
  <Modal.Title id={"user" + user.id}>
    {user.lastName}, {user.firstName}
  </Modal.Title>
</Modal.Header>
<Modal.Body>
  <Row>
    <Col>Status</Col>
    <Col>{user.activeStatus}</Col>
  </Row>
  <Row>
    <Col>Role</Col>
    <Col>{user.role}</Col>
  </Row>
</Modal.Body>
</Modal>
```

- Works well with React
- Readable
- Can be paired with traditional bootstrap plugin

`<div class="modal modal-header">` → `<Modal.Header>`



React-Bootstrap site vs Bootstrap site

DAILY BEHAVIOR

- 3 functions are run daily
- Node-cron library
- Javascript dates are tricky!

```
const nodeCron = require("node-cron");
```

```
const today = new Date().setUTCHours(0, 0, 0, 0);  
const dbDate = new Date(account.nextPaymentDueDate).setUTCHours(0, 0, 0, 0);
```

```
function cron() {  
  nodeCron.schedule("0 8 * * *", async function () {  
    console.log("daily at 8");  
  
    checkWarnings();  
    checkRecurringPayments();  
    checkCreditUpdates();  
  });  
}
```

```
const today = new Date().setUTCHours(0, 0, 0, 0);  
const dbDate = new Date(account.nextPaymentDueDate).setUTCHours(0, 0, 0, 0);
```


PAGINATION

- jw-paginate library for backend pagination
- Functional component to determine and render links
- useState with onClick event and useEffect fetches update

```
const paginate = require('jw-paginate');
```

```
const page = parseInt(req.query.page) || 1;
const pageSize = 4;
const pager = paginate(chequeList[0].length, page, pageSize);
const pageOfCheques = chequeList[0].slice(pager.startIndex, pager.endIndex + 1);

return res.json({pager, pageOfCheques});
})
```

```
const [paginate, setPaginate] = useState("");
const [page, setPage] = useState(0);

function RenderPages() {
  if(!paginate || !paginate.currentPage){
    return <p>No cheques for approval</p>
  }
  let pages = [];
  if (paginate.currentPage > 1) {
    pages.push(<li className="page-item"><a className="page-link" onClick={() => {setPage(paginate.currentPage - 1);}}>Pr
  } else {
    pages.push( <li className="page-item disabled"><a className="page-link " onClick={() => {setPage(paginate.currentPage
  }
  for (let i = 1; i < paginate.totalPages + 1; i++) {
    if (i === paginate.currentPage) {
      pages.push(<li className="page-item active"><a className="page-link" onClick={() => {setPage(i);}}>{i}</a></li> );
    } else {
      pages.push(<li className="page-item"><a className="page-link" onClick={() => {setPage(i);}}>{i}</a> </li>);
    }
  }
}
```

```
    </ListGroup>
    <RenderPages />
  </Card>
```

FILE DISPLAY

- Challenge: Display cheque images without base64 encoding
- Solution: Return buffer with content type image and it will be displayed as an image

```
res.header("Content-Type", "image/jpeg").send(result.Body);
```

Transaction # 4



Payee name:	Kate Post
Upload date:	2022-01-24
Value on cheque:	56.31
Payer account number	44
Cheque number	5673

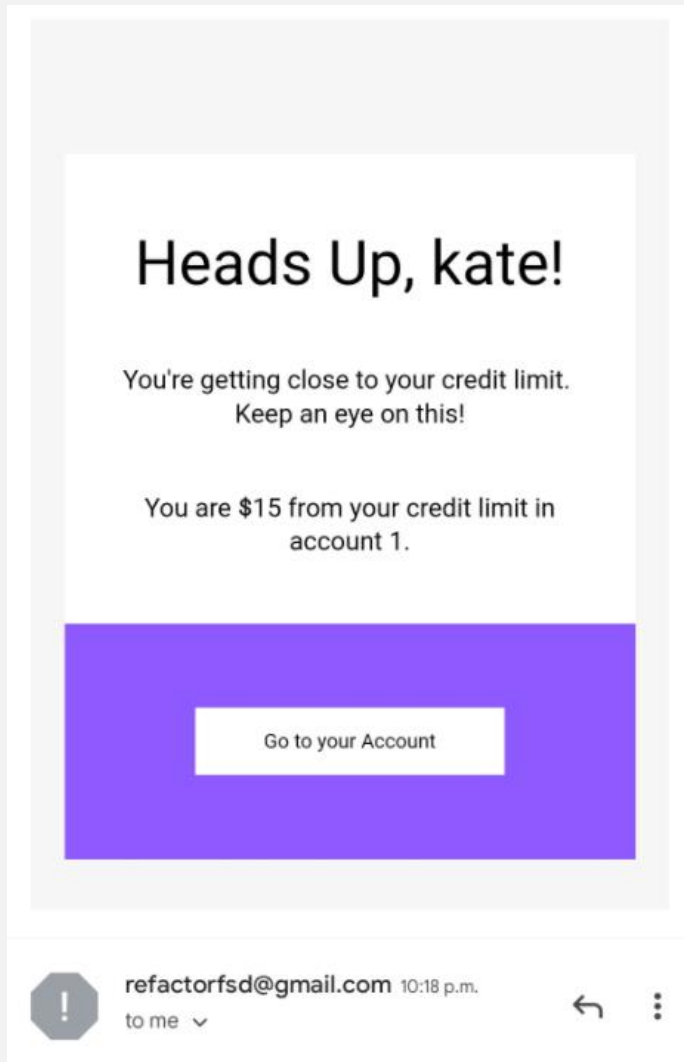
[Approve Cheque](#) [Deny Cheque](#)

```
<img src={`http://localhost:3001/admins/chequeImage/${chequeTransaction.chequeId}/${token}`} width="700px" alt="cheque"></img>  
Col>
```

COMMUNICATION

Challenge: Alert users to changes in their accounts

Solution: Automated emails



```
function email(to, firstName, lastName, message, warning) {
  console.log({
    from: process.env.SENDGRID_SENDER_EMAIL,
    template: process.env.SENDGRID_TEMPLATE,
  });
  const msg = {
    to: to,
    name: firstName + " " + lastName,
    from: process.env.SENDGRID_SENDER_EMAIL,
    template_id: process.env.SENDGRID_TEMPLATE,
    dynamic_template_data: {
      user: firstName,
      message: message,
      warning: warning,
    },
  };
  sgMail
    .send(msg)
    .then((response) => {
      console.log(response[0].statusCode);
      console.log(response[0].headers);
    })
    .catch((error) => {
      console.error(error);
    });
}
```



WHAT WE LEARNED



CONTEXTS

- Provide a method of storing and accessing information globally
- Any descendent of the context provider has access to the context
- Eliminates the need to pass variables down through the line of descent

```
import { createContext } from "react";

export const AuthContext = createContext({
  authState: null,
  authUser: {},
  setAuthState: null,
  setAuthUser: null
});
```

```
return (
  <AuthContext.Provider value={{authState, setAuthState, authUser, setAuthUser}}>
    {children}
  </AuthContext.Provider>
)
}
export default GetAuth;
```

```
return (
  <div className="App">
    <GetAuth>
    <Router>
      <Routes>
```

```
function Client() {
  const state = useContext(AuthContext);
  if(state.authState === false){
    return <Navigate to='/login'/>
  }
  if(state.authState === true){
    if(state.authUser.role === 'admin'){
      return <Navigate to='/admin'/>
    }
    if(state.authUser.role === 'client'){
      switch(state.authUser.status){
        case 'disabled':
          return <Navigate to='/login'/>
        case 'inactive':
          return <div>Sit tight!<br/>You
        case 'active':
          return <Outlet />
        default:
          break;
      }
    }
  }
  return <div>Loading...</div>
}
```

HEROKU DEPLOYMENT

```

  build
  static
    css
      # main.d8324f1a.css
      # main.d8324f1a.css.map
    js
      JS main.c0752642.js
      ≡ main.c0752642.js.LICENSE.txt
      JS main.c0752642.js.map
  media
    loading.84ac76a547ae9c30b3dd.gif
    {} asset-manifest.json
    favicon.png
    index.html
    {} manifest.json
    refactorlogo.png
    robots.txt

```

```

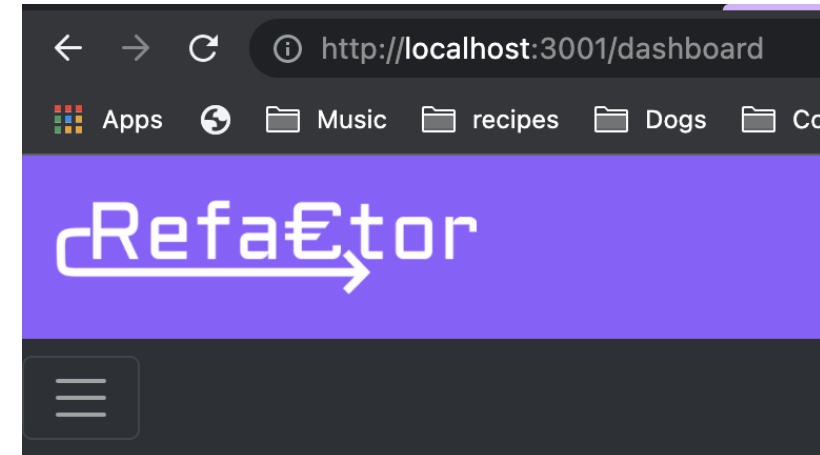
  server
    build
    config
      JS config.js
      dev.env
    misc
      JS authware.js
      JS cron.js
    models
    node_modules
    routes
    utils
    .gitignore
    JS index.js
    {} package-lock.json
    {} package.json

```

```

node_modules
dev.env
package-lock.json
.DS_Store

```



```

app.use("/transactions", transactionsRouter);
app.use("/users", usersRouter);
app.use("/admins", adminsRouter);
app.use("/accounts", accountsRouter);

app.use(express.static("./build"));

app.get("/*", function (req, res) {
  res.sendFile(path.resolve(__dirname, 'build', 'index.html'));
})

```

HEROKU DEPLOYMENT

```
heroku login
cd /Users/aly/Documents/JohnAbbott/refactor/server
git init

heroku git:remote -a refactor-banking
git add .
git commit -am "make it better"
git push heroku master
```

username password host

mysql://b9f34266ab468b:cdabe46a@us-cdbr-east-05.cleardb.net/heroku_6b792346acddeda?reconnect=true

```
heroku run bash -a refactor-banking
```

← Access Heroku hosted directories

PROMISES

Async functions always return a promise. If the return value of an async function is not explicitly a promise, it will be implicitly wrapped in a promise.

For example, the following:

```
async function foo() {  
  return 1  
}
```

...is similar to:

```
function foo() {  
  return Promise.resolve(1)  
}
```

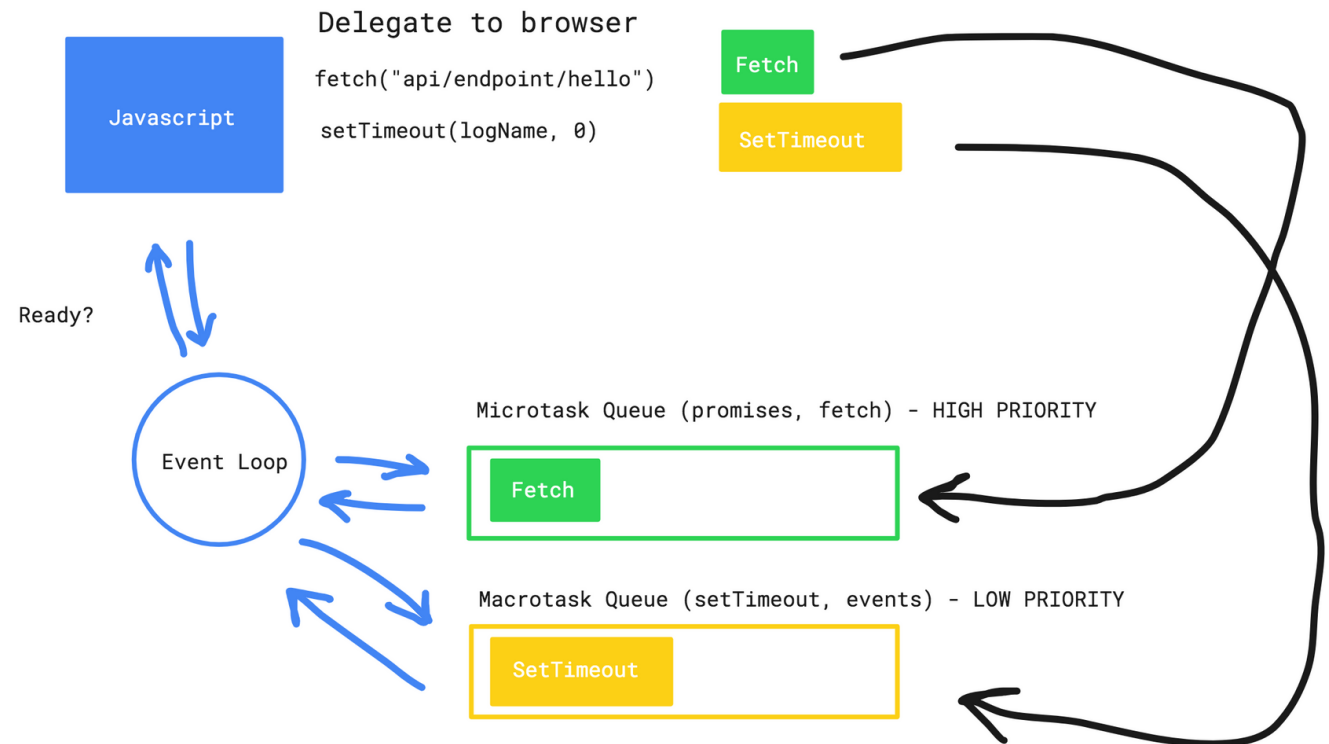


Image source: <https://www.freecodecamp.org/news/async-await-javascript-tutorial/>

PROMISES

```
function downloadCheque(chèqueId) {
  return new Promise((resolve, reject) => {
    Cheques.findByPk(chèqueId)
      .then((result1) => {
        downloadFromS3(result1.s3key)
          .then((result2) => {
            resolve(result2);
          })
          .catch((error) => {
            reject(error);
          });
      })
      .catch((err) => {
        reject(err);
      });
  });
}

const downloadFromS3 = (key) => {
  const downloadParams = {
    Bucket: process.env.S3BUCKET,
    Key: key,
  };
  return s3.getObject(downloadParams).promise();
};
```

- Custom promise
- s3.getObject callback as inline promise
- .then and .catch on function returning new Promise

```
downloadCheque(chèqueId)
  .then((result) => {
    res.header("Content-Type", "image/jpeg").send(result.Body);
  })
  .catch((err) => {
    console.log(err);
    res.status(400).json(err);
  });
```

FILE STORAGE

```
await axios
  .post(`/transactions/depositCheque`, formData, {
    headers: {
      "Content-Type": "multipart/form-data",
      accessToken: localStorage.getItem("accessToken"),
    },
  })
  .then((result) => {
    console.log(result.data);
    setSuccess("Cheque deposited, awaiting approval.");
  })
  .catch((err) => {
    setAlert(err.response.data);
    console.log(err.response.data);
  });
setLoading(false);
```

```
const uploadToS3 = (buffer, name, type) => {
  const uploadParams = {
    Bucket: process.env.S3BUCKET,
    Key: name + "." + type,
    Body: buffer,
  };
  return s3.upload(uploadParams).promise();
};
```

```
const multipart = require("multipart");
```

```
form.parse(req, async (error, fields, files) => {
  if (error) {
    reject(error);
  }
  try {
    const path = files.file[0].path;
    if (!path) reject("No file uploaded");
    const body = { path: path, fields: fields };
    resolve(body);
  } catch (err) {
    reject(err);
  }
});
```

Deposit a cheque

Account	Account number on Cheque
<input type="text" value="Account #4 - CAD debit"/>	<input type="text" value="44"/>
Value	Cheque S/N
<input type="text" value="13.23"/>	<input type="text" value="25"/>
Upload image	
<input type="button" value="Choose File"/> <input type="text" value="cheque.jpg"/>	
<input type="button" value="Send"/>	



FUTURE WORK

REFACTOR



FUTURE WORK

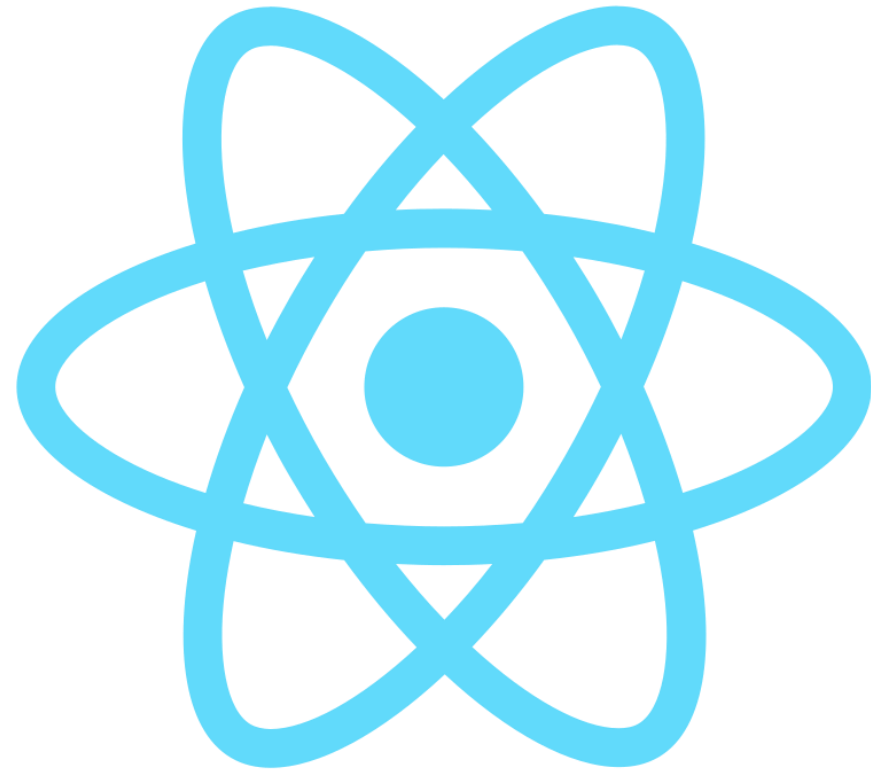
- Built in communication system for automatic correspondence with clients
- Implementing the cash back system on the credit accounts
- Tools like interest / mortgage calculators
- ATM locator

SUMMARY

REFACTOR

REACT

- Rich, powerful
- Stunning static web pages
- Asynchronous
- Steep learning curve





THANK YOU

REFACTOR