Web Development Ⅱ Final Project

Masahiro (Leader)

Taisei

Bosco

Contents

Task Division

Masahiro

- Common Functions Part
- Category Part
- Transaction Validation
- Toast Notification
- Login Part

Taisei

- Transaction Part
- Filter Part
- Confirmation Message
- Calculate Account Balance
- Responsive Design

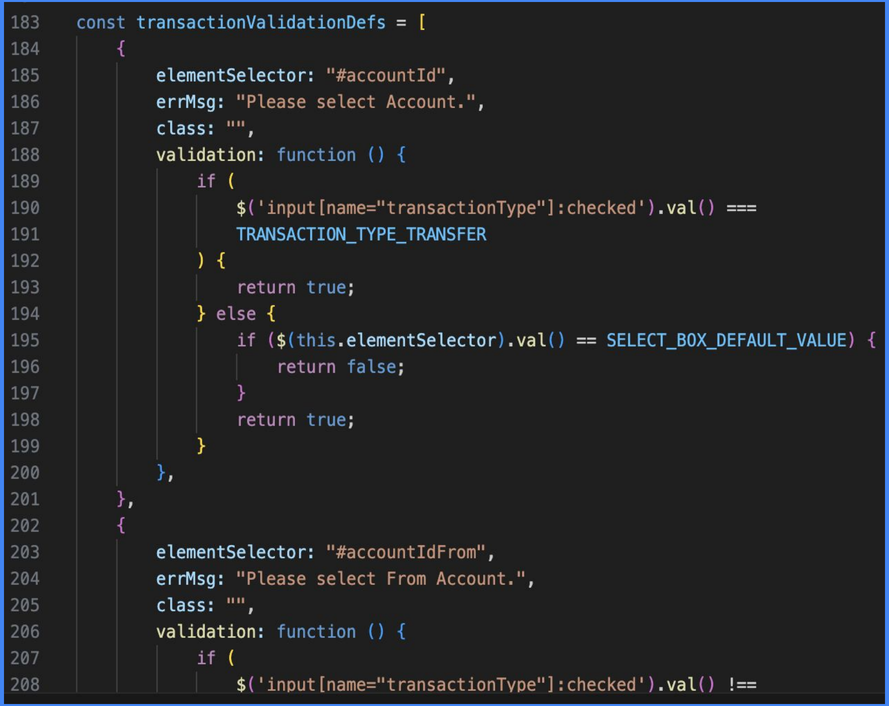
Bosco

- Account Part
- CSS Styling
- Create Web App Design
- Duplicate alert

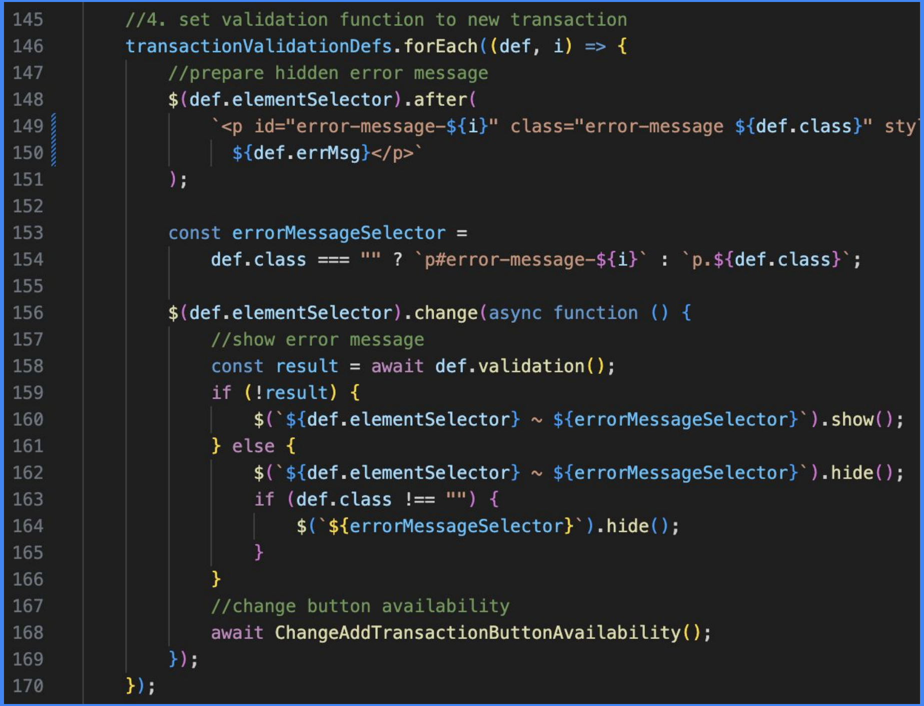
Big Issues

Big Issues - Masahiro

Handling validations for creating new transaction.



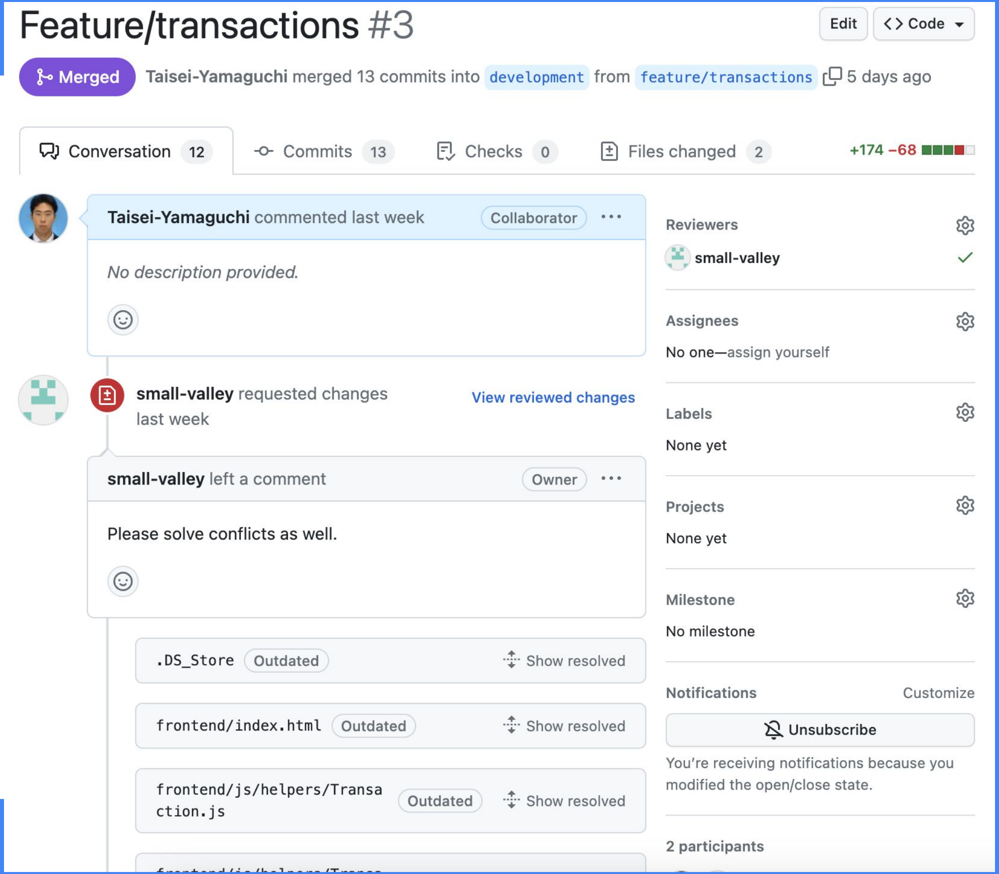
```
const transactionValidationDefs = [
    {
        elementSelector: "#accountId",
        errMsg: "Please select Account.",
        class: "",
        validation: function () {
            if (
                $('input[name="transactionType"]:checked').val() ===
                TRANSACTION_TYPE_TRANSFER
            ) {
                return true;
            } else {
                if ($(this.elementSelector).val() == SELECT_BOX_DEFAULT_VALUE) {
                    return false;
                }
                return true;
            }
        },
    },
    {
        elementSelector: "#accountIdFrom",
        errMsg: "Please select From Account.",
        class: "",
        validation: function () {
            if (
                $('input[name="transactionType"]:checked').val() !==
```



```
    //4. set validation function to new transaction
    transactionValidationDefs.forEach((def, i) => {
        //prepare hidden error message
        $(def.elementSelector).after(
            `<p id="error-message-${i}" class="error-message ${def.class}" sty
             ${def.errMsg}</p>`
        );

        const errorMessageSelector =
            def.class === "" ? `p#error-message-${i}` : `p.${def.class}`;

        $(def.elementSelector).change(async function () {
            //show error message
            const result = await def.validation();
            if (!result) {
                $(`${def.elementSelector} ~ ${errorMessageSelector}`).show();
            } else {
                $(`${def.elementSelector} ~ ${errorMessageSelector}`).hide();
                if (def.class !== "") {
                    $(`${errorMessageSelector}`).hide();
                }
            }
            //change button availability
            await ChangeAddTransactionButtonAvailability();
        });
    });
```

Big Issues - Taisei

- Unwilling commits can make merging difficult
- Coordination of commits among team members is important
- Pay attention to commits for project success


Feature/transactions #3

Edit | <> Code

Merged Taisei-Yamaguchi merged 13 commits into development from feature/transactions 5 days ago

Conversation 12 | Commits 13 | Checks 0 | Files changed 2 | +174 −68

Taisei-Yamaguchi commented last week Collaborator

No description provided.

small-valley requested changes last week View reviewed changes

small-valley left a comment Owner

Please solve conflicts as well.

.DS_Store Outdated Show resolved

frontend/index.html Outdated Show resolved

frontend/js/helpers/Transaction.js Outdated Show resolved

Reviewers: small-valley ✓

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications Customize

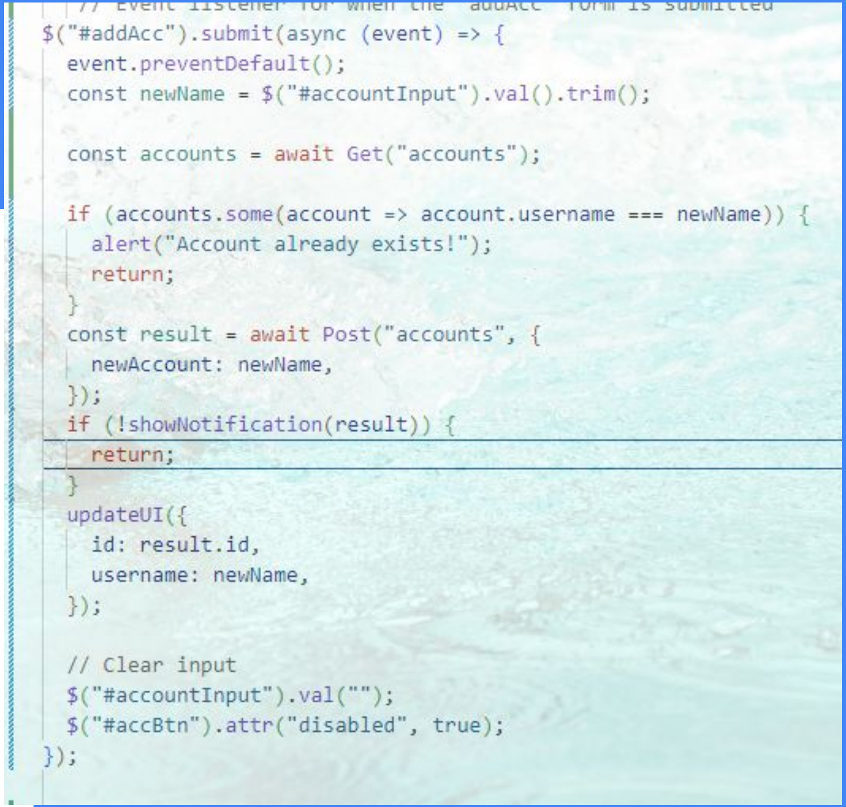
Unsubscribe

You're receiving notifications because you modified the open/close state.

2 participants

Big Issues - Bosco

- using the wrong logic to add the account to the server
- failed to get wanted information from server

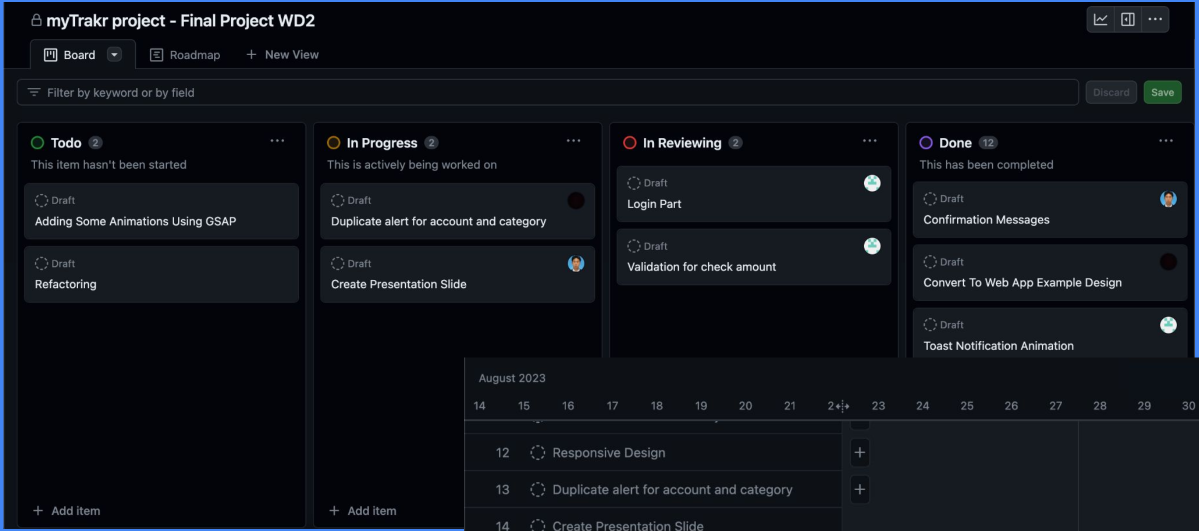


```
  // Event listener for when the "addAcc" form is submitted
$("#addAcc").submit(async (event) => {
  event.preventDefault();
  const newName = $("#accountInput").val().trim();

  const accounts = await Get("accounts");

  if (accounts.some(account => account.username === newName)) {
    alert("Account already exists!");
    return;
  }
  const result = await Post("accounts", {
    newAccount: newName,
  });
  if (!showNotification(result)) {
    return;
  }
  updateUI({
    id: result.id,
    username: newName,
  });

  // Clear input
  $("#accountInput").val("");
  $("#accBtn").attr("disabled", true);
});
```

What We Learned

What We Learned - Masahiro

Using GitHub Projects to manage our project.



myTrakr project - Final Project WD2

Board | Roadmap | New View

Filter by keyword or by field

Discard | Save

Todo 2
This item hasn't been started
- Draft: Adding Some Animations Using GSAP
- Draft: Refactoring

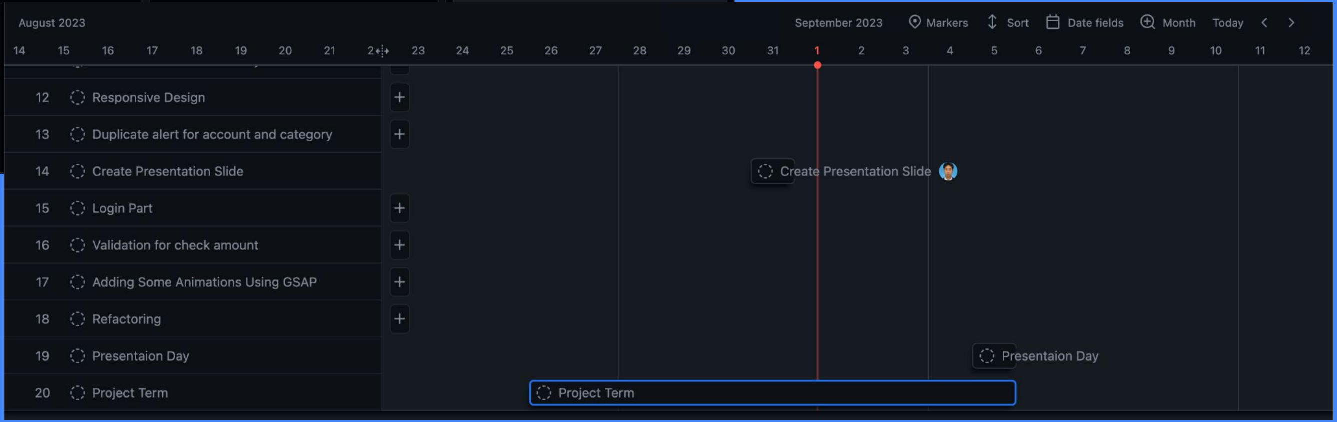
Add item

In Progress 2
This is actively being worked on
- Draft: Duplicate alert for account and category
- Draft: Create Presentation Slide

Add item

In Reviewing 2
- Draft: Login Part
- Draft: Validation for check amount

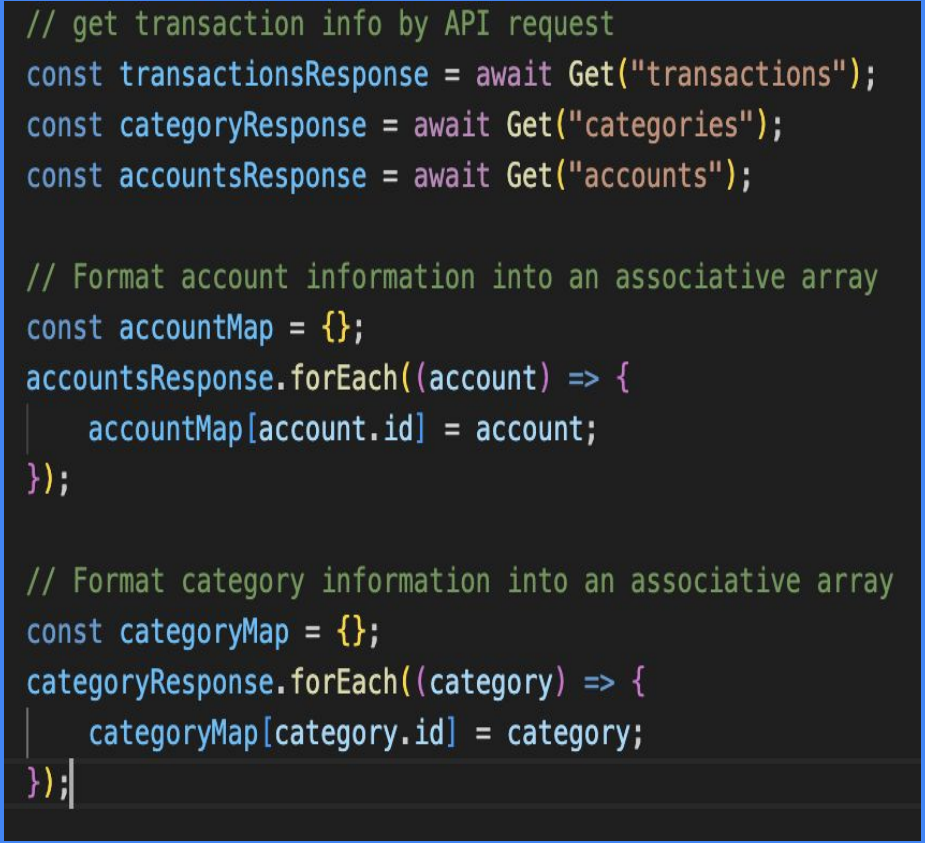
Done 12
This has been completed
- Draft: Confirmation Messages
- Draft: Convert To Web App Example Design
- Draft: Toast Notification Animation



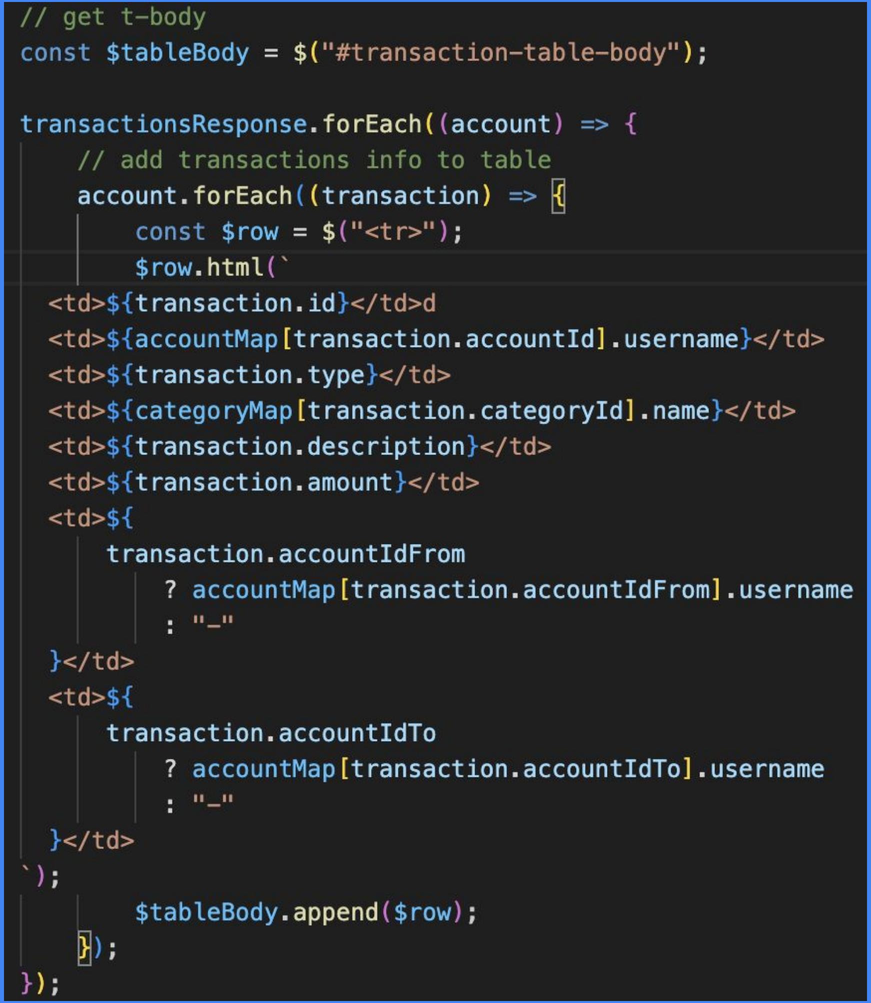
August 2023 | September 2023 | Markers | Sort | Date fields | Month | Today

12	Responsive Design
13	Duplicate alert for account and category
14	Create Presentation Slide
15	Login Part
16	Validation for check amount
17	Adding Some Animations Using GSAP
18	Refactoring
19	Presentaion Day
20	Project Term

What We Learned - Taisei



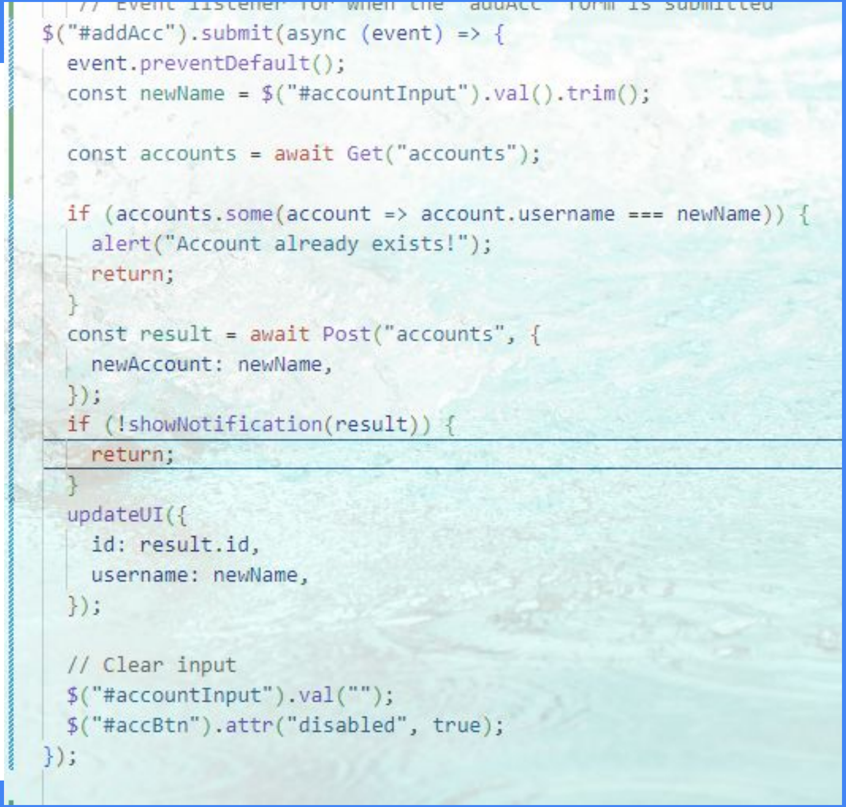
```javascript
// get transaction info by API request
const transactionsResponse = await Get("transactions");
const categoryResponse = await Get("categories");
const accountsResponse = await Get("accounts");

// Format account information into an associative array
const accountMap = {};
accountsResponse.forEach((account) => {
    accountMap[account.id] = account;
});

// Format category information into an associative array
const categoryMap = {};
categoryResponse.forEach((category) => {
    categoryMap[category.id] = category;
});
```



```javascript
// get t-body
const $tableBody = $("#transaction-table-body");

transactionsResponse.forEach((account) => {
    // add transactions info to table
    account.forEach((transaction) => {
        const $row = $("<tr>");
        $row.html(`
  <td>${transaction.id}</td>d
  <td>${accountMap[transaction.accountId].username}</td>
  <td>${transaction.type}</td>
  <td>${categoryMap[transaction.categoryId].name}</td>
  <td>${transaction.description}</td>
  <td>${transaction.amount}</td>
  <td>${
    transaction.accountIdFrom
      ? accountMap[transaction.accountIdFrom].username
      : "-"
  }</td>
  <td>${
    transaction.accountIdTo
      ? accountMap[transaction.accountIdTo].username
      : "-"
  }</td>
`);
        $tableBody.append($row);
    });
});
```

What We Learned - Bosco

- getting and posting information from server
- compare data from server with user input



```
  // Event listener for when the "addAcc" form is submitted
$("#addAcc").submit(async (event) => {
  event.preventDefault();
  const newName = $("#accountInput").val().trim();

  const accounts = await Get("accounts");

  if (accounts.some(account => account.username === newName)) {
    alert("Account already exists!");
    return;
  }
  const result = await Post("accounts", {
    newAccount: newName,
  });
  if (!showNotification(result)) {
    return;
  }
  updateUI({
    id: result.id,
    username: newName,
  });

  // Clear input
  $("#accountInput").val("");
  $("#accBtn").attr("disabled", true);
});
```

Future Improvements

- Add some animations for better UI.
- Code refactoring, Styling and Designing
- Transforming all html codes into jQuery

Demonstration