**SW Engineering CSC 648/848**

**Section 04 Team 01**

**KeepUp App**

**Milestone 4 Document**

**3rd May, 2023**

**1A)QA Testing - Unit Testing:**

KeepUp app can be classified into following main sections: User registration ; User Login ; Notepad ; Expense Tracker ; Relaxer

We have performed the following unit tests to cover the P1 features:

Expense Tracker – Expenses:

The expenses in the Expense Tracker section of the app helps the user to keep a track of all their finances on a daily basis.

**P1 Feature 1:** Adding expense : Enables the user to add an expense by providing the required input details. [Defined by addExpense function shown in the picture below]

**P1 Feature 2:** Displaying all Expenses: Enables users to view all the expenses they added. [Defined by getExpense function shown in the picture below]

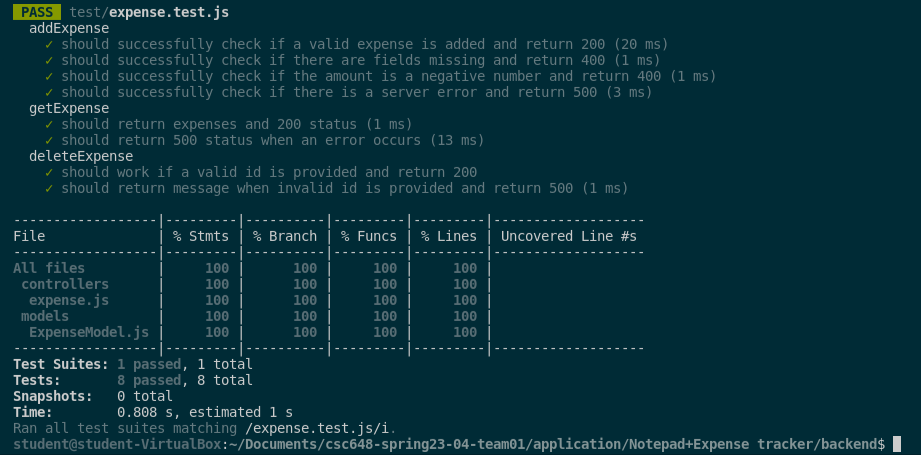
**P1 Feature 3:** Deletion of expense: Enables user to delete any expense they added. [Defined by deleteExpense function shown in the picture below]

Corresponding source files:

* master/application/ Notes+Expense tracker/backend/controllers/expense.js
* master/application/ Notes+Expense tracker/backend/models/ExpenseModel.js

Test case directory:

* master/application/ Notes+Expense tracker/backend/test/expense.test.js



When testing the expenses feature, the unit tests cover 100% of the functions and 100% of the statements. All three P1 functions, addExpense, getExpense, and deleteExpense, ideal scenarios are correctly working as intended and the test also does error checking if something were to break in these functions.

Expense Tracker – Incomes:

The incomes in the Expense Tracker section of the app helps the user visualize the amount of income they are making.

**P1 Feature 4:** Adding income : Enables the user to add an income by providing the required input details. [Defined by addIncome function shown in the picture below]

**P1 Feature 5:** Displaying all Incomes: Enables users to view all the incomes they added. [Defined by getIncome function shown in the picture below]

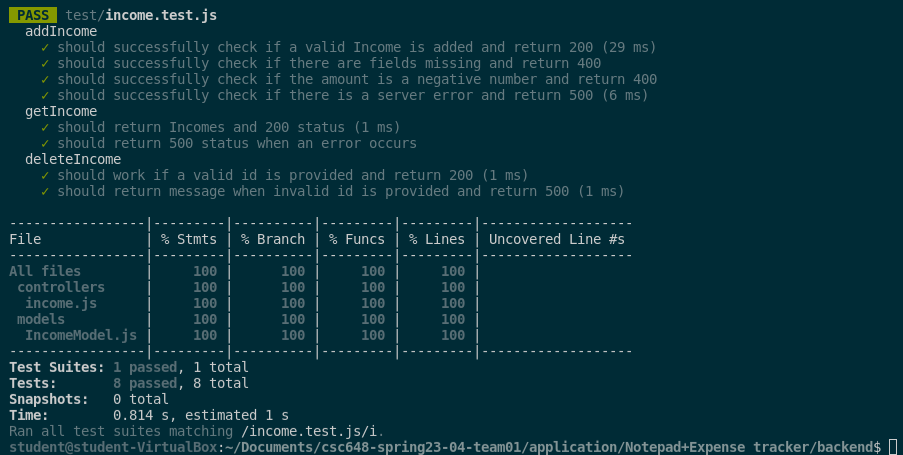
**P1 Feature 6:** Deletion of income: Enables user to delete any income they added. [Defined by deleteIncome function shown in the picture below]

Corresponding source files:

* master/application/Notes+Expense tracker/backend/controllers/income.js
* master/application/ Notes+Expense tracker/backend/models/IncomeModel.js

Test case directory:

* master/application/Notes+Expense tracker /backend/test/income.test.js



When testing the three P1 functions of Income, 100% of the function coverage and 100% of the statement coverage have been successfully tested without any issues. The functions addIncome, getIncome, and deleteIncome properly works in an ideal scenario and the test also does error checking if something were to break in these functions.

Notepad:

**P1 Feature 7:** Display all notes : User must be able to view all the notes he added on the home page once he logs in. [Defined by “Get note” function shown in the picture below]

**P1 Feature 8:** Create notes: Enables users to add new notes. [Defined by “Create note” function shown in the picture below]

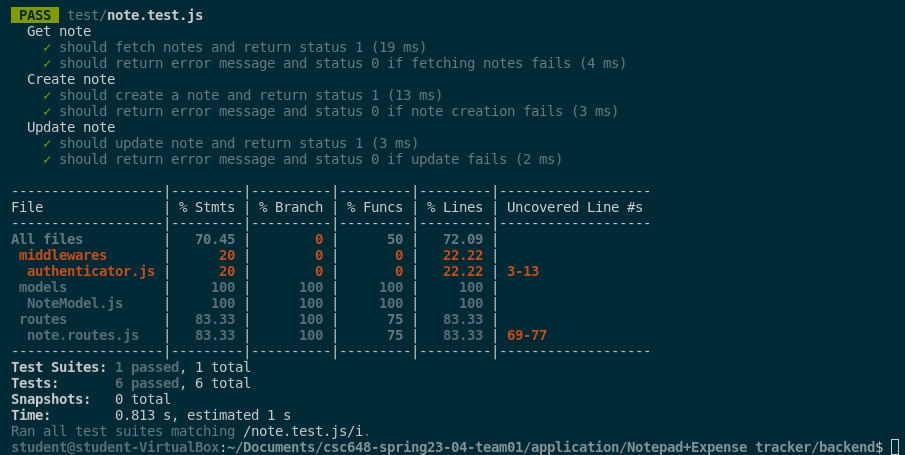
**P1 Feature 9:** Update notes : Enables the user to make changes to existing notes ny updating it. [Defined by “Update note” function shown in the picture below]

Corresponding source files:

* master/application/Notes+Expense tracker/backend/ models/NoteModel.js
* master/application/ Notes+Expense tracker/backend/routes/note.routes.js

Test case directory:

* master/application/Notes+Expense tracker /backend/test/note.test.js



**Note: The uncovered lines 69-77 shown above are that of the “delete notes” feature which we did not test here.**

When testing the P1 features display, add and update notes, the functional coverage is 100% and statement coverage is 100% as well. The functions Get note, Create note, and Update note properly work in an ideal scenario, and the test also does error checking if something were to break in these functions.

User Registration:

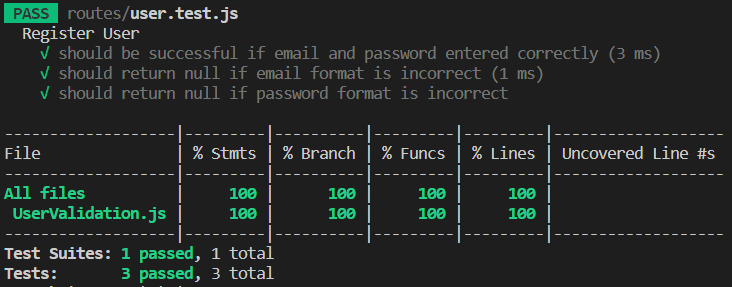
**P1 Feature 10:** User registration : Enables the user to register with email and password. [Defined by “RegisterUser” function shown in the picture below]

Corresponding source files:

* master/application/ Notes+Expense tracker/backend/routes/UserValidation.js

Test case directory:

* master/application/Notes+Expense tracker /backend/test/user.test.js



The results show that 100% statement and functional coverage for the P1 feature. It checks for the valid formats of email id and password the user provides while registering.

**1B) Integration Testing:**

**Feature 1: User Registration** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_001 | Test Case Discription | Testing the Signup functionaltiy | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | Review from Professor, Incorporated in cversion 1.1 | |  |  |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Name = Malavya test** |  |
| **2** |  |  | **2** | **Email-ID = malavyatest@gmail.com** | |
| **3** |  |  | **3** | **PassWord = testing123** |  |
|  |  |  |  |  |  |
| Test Scenario | **Creating a Account by valid name, Email-ID and Password** | |  |  |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/ | Site should Open | As Expected | PASS |  |
| 2 | Press on Sign up | Sign up page is opened | As Expected | PASS |  |
| 3 | Enter Name, Email-ID and Password | Credentials can be entered | As Expected | PASS |  |
| 4 | Click Sign up | User has created an account | As Expected | PASS |  |

**Feature 2: User Login** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_002 | Test Case Discription | Testing the Login functionaltiy | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | Review from Professor, Incorporated in cversion 1.1 | |  |  |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Email-ID = malavyatest@gmail.com** |  |
| **2** | **Created Account** |  | **2** | **PassWord = testing123** |  |
|  |  |  |  |  |  |
| Test Scenario | **Verifying the Email-ID and the password, the user can log in** | |  |  |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/ | Site should Open | As Expected | PASS |  |
| 2 | Press on Login | Login page is opened | As Expected | PASS |  |
| 3 | Enter Email-ID and Password | Credentials can be entered | As Expected | PASS |  |
| 4 | Click Log in | User has loged in to the account | As Expected | PASS |  |

**Feature 3: Add notes** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_003 | Test Case Discription | Testing the Adding notes | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Title = Test1** |  |
| **2** | **Created Account** |  | **2** | **Description = Test notes** |  |
|  |  |  |  |  |  |
| Test Scenario | | **A note should be created, a note is added** | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/notes | Site should Open with the notes page | As Expected | PASS |  |
| 2 | Press on "+" on bottom left | Create new note option appears | As Expected | PASS |  |
| 3 | Enter Title and description of the note | Title and description could be entered | As Expected | PASS |  |
| 4 | Click Create | Note is created | As Expected | PASS |  |

**Feature 4: Update notes** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_004 | Test Case Discription | Testing the Update notes feature | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Title = Test1Upd** |  |
| **2** | **Created Account** |  | **2** | **Description = Test NoteUpd** |  |
| **3** | **A note present in the page** |  |  |  |  |
|  |  |  |  |  |  |
| Test Scenario | | A note already present can be modified, the note is updated | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/notes | Site should Open | As Expected | PASS |  |
| 2 | Press on Update in existing notes | The note opens and changes can be made | As Expected | PASS |  |
| 3 | Click Update | The note is now updated | As Expected | PASS |  |

**Feature 5: Delete notes** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_005 | Test Case Discription | Testing the Deletion of Notes | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Title = Test1Upd** |  |
| **2** | **Logged-In account** |  | **2** | **Description = Test NoteUpd** |  |
| **3** | **A note present in the page** |  |  |  |  |
|  |  |  |  |  |  |
| Test Scenario | | Any existing note selected should be deleted, the note is deleted | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/notes | Site should Open | As Expected | PASS |  |
| 2 | Press on Delete button on the note we want to deleted | The note is deleted | As Expected | PASS |  |

**Feature 6: Add income** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_006 | Test Case Discription | Testing the Additoin of Income | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Salary Title = Software Enigneering Job** |  |
| **2** | **Created Account** |  | **2** | **Salary Amount = 10000** |  |
|  |  |  | **3** | **Date = 01/05/2023** |  |
|  |  |  | **4** | **Option = Salary** |  |
|  |  |  | **5** | **Reference = Salary, recived with bonus** |  |
|  |  |  |  |  |  |
| Test Scenario | | The Income should be added and displayed | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/expenses | Expense page is opened | As Expected | PASS |  |
| 3 | Press on Incomes | Incomes page is opened | As Expected | PASS |  |
| 4 | Enter all input | The data can be inserted | As Expected | PASS |  |
| 5 | Click add Income | Income is added and displayed | As Expected | PASS |  |

**Feature 7: Delete income** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_007 | Test Case Discription | Testing the Income Deletion Functionality | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Salary Title = Software Enigneering Job** |  |
| **2** | **Created Account** |  | **2** | **Salary Amount = 10000** |  |
| **3** | **Income Present in the account** |  | **3** | **Date = 01/05/2023** |  |
|  |  |  | **4** | **Option = Salary** |  |
|  |  |  | **5** | **Reference = Salary, recived with bonus** |  |
|  |  |  |  |  |  |
| Test Scenario | | The Income should be Deleted and not displayed | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/expenses | Site should Open | As Expected | PASS |  |
| 2 | Press on The delete Icon for the income that needs to be deleted | The income is deleted | As Expected | PASS |  |

**Feature 8: Add expense** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_008 | Test Case Discription | Testing the Expenses Addition functionaltiy | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Expense Title = Target Shopping** |  |
| **2** | **Created Account** |  | **2** | **Expense Amount = 100** |  |
|  |  |  | **3** | **Date = 02/05/2023** |  |
|  |  |  | **4** | **Option = Groceries** |  |
|  |  |  | **5** | **Reference = Noodles, Bread and eggs** |  |
|  |  |  |  |  |  |
| Test Scenario | | The expenses should be added and displayed | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/expenses | Site should Open | As Expected | PASS |  |
| 2 | Press on Expenses | Expenses page is opened | As Expected | PASS |  |
| 3 | Enter all inputs | The data can be inserted | As Expected | PASS |  |
| 4 | Click in Add Expenses | Expense is added and displayed | As Expected | PASS |  |

**Feature 9: Delete expense** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_009 | Test Case Discription | Testing the Expenses deletion functionaltiy | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | PASS |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Expense Title = Target Shopping** |  |
| **2** | **Created Account** |  | **2** | **Expense Amount = 100** |  |
| **3** | **Existing Expense in page** |  | **3** | **Date = 02/05/2023** |  |
|  |  |  | **4** | **Option = Groceries** |  |
|  |  |  | **5** | **Reference = Noodles, Bread and eggs** |  |
|  |  |  |  |  |  |
| Test Scenario | | **The expenses should be Deleted from page** | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/expenses | Site should Open | As Expected | PASS |  |
| 2 | Press on The delete Icon for the expenses to be deleted | Expense is deleted | As Expected | PASS |  |

**Feature 10: Breather** [Fail]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_010 | Test Case Discription | Testing the Breating functionaltiy | |  |
| Created By | Malavya | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Malavya | Date Tested | 01-May-23 | Test Case | FAIL |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Email-ID = malavyatest@gmail.com** |  |
| **2** | **Created Account** |  | **2** | **PassWord = testing123** |  |
|  |  |  |  |  |  |
| Test Scenario | | **The breather should function** | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/breathe | Site should Open | As Expected | PASS |  |
| 2 | Press on start | Breather graphics should start | Breather displayed but not moving | FAIL |  |

**Feature 11: Relaxation Exercises** [Pass]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | BU\_011 | Test Case Discription | Testing the exercises display | |  |
| Created By | Hruthika | Reviewed By | Hruthika | Version | 1.1 |
|  |  |  |  |  |  |
| QA Tester's Log | | Review from Professor, Incorporated in cversion 1.1 | | |  |
|  |  |  |  |  |  |
| Teseter's Name | Hruthika | Date Tested | 01-May-23 | Test Case | Pass |
|  |  |  |  |  |  |
| **S#** | **Prerequisite** |  | **S#** | **Test Data** |  |
| **1** | **Chrome Access** |  | **1** | **Email-ID = malavyatest@gmail.com** |  |
| **2** | **Created Account** |  | **2** | **PassWord = testing123** |  |
|  |  |  |  |  |  |
| Test Scenario | | **The relaxation exercises should be displayed** | | |  |
|  |  |  |  |  |  |
| Step # | Step Details | Expected Results | Actual Results | Pass/Fail/Not Executed/Suspended |  |
| 1 | Navigate to http://34.102.4.227/breathe | Site should Open | As Expected | PASS |  |
| 2 | Scroll down to view ralaxation exercises | Exercises should be displayed | As Expected | PASS |  |

10 of the P1 features have passed the integration testing. One feature failed because the relaxer app is still in progress and the front-end lead is working on it. It will be completed by M5.

**URL for issue tracker:** https://github.com/CSC-648-SFSU/csc648-spring23-04-team01/issues

**2. Coding Practices:**

We are practicing the use of **WordPress JS coding standards** in out app development to ensure that the code looks like a single-person typed in, though multiple people worked on it.

We are using Prettier – Code Formatter [It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary] and JSHint to ensure that the coding style is being followed. Also, verification of certain practices like folder organization are being verified by the Frontend lead.

Some coding practices we strictly enforced are:

1. Adding all import statements only at top of each page (Shouldn’t write import statement before the function that user it in middle of the page)
2. Comments should be added before each function, describing the purpose of it.
3. Comments should be added at beginning of important pages to specify the main functionality of that page
4. Naming of files: Single word => Can start with either lower or uppercase followed by all lower case

letters [Ex: user.js or User.js]

Multiple wording => 1st word can start with upper or lower case, followed by each

word must start with capital and no spaces. All intermediate words must be small.

[Ex: IncomePage.js or incomePaje.js]

1. All unit test source code files should be under one folder.
2. One js file should contain one functionality at a time.
3. All images etc should be under one folder.
4. All routes should be under one folder
5. All the backend code should be in the backend folder and all frontend code in frontend folder only.

List of source files for coding style:

1. master/application/Notepad+Expense tracker/backend/routes/noteRoutes.js
2. master/application/Notepad+Expense tracker/backend/routes/userRoutes.js
3. master/application/Notepad+Expense tracker/backend/routes/transactions.js

master/application/Notepad+Expense tracker/backend/controllers/expense.js

1. master/application/Notepad+Expense tracker/backend/controllers/income.js
2. master/application/Notepad+Expense tracker/frontend/src/pages/ExpensePage.jsx

master/application/Notepad+Expense tracker/frontend/src/components/Dashboard/Dashboard.js

master/application/Notepad+Expense tracker/frontend/src/components/Expenses/Expenses.js

master/application/Notepad+Expense tracker/frontend/src/components/Income/Income.js

master/application/Notepad+Expense tracker/frontend/src/components/Form/Form.js