Source Distribution Release Notes

Dam Dollars – Source Distribution

These release notes describe the steps required to build, test, and further develop Dam Dollars from the source code. This document is intended for developers picking up the project and includes detailed instructions on obtaining the source code, building the application, running tests, and further development.

---

1. Obtaining the Source Code

- Clone the repository using:

  ```
  git clone https://github.com/henderos/CS362-Class-Project.git
  cd 362-Class-Project
  ```

---

2. Directory Structure

The project is organized as follows:

```
CS362-Class-Project/
├── personal finance analyzer/      # Source code
│   ├── backend/
│   │   ├── src/
│   │   │   ├── index.js
│   │   │   ├── plaidtest.html
│   │   │   ├── transactions.json
│   │   │   ├── package.json
│   │   │   ├── package-lock.json
│   │   │   └── server.js
│   │   ├── tests/  #backend tests
│   ├── frontend/
│   │   ├── public/
│   │   │   ├── css/
│   │   │   └── images/
│   │   ├── index.js
│   │   ├── login.html
│   │   ├── signup.html
```

```
│   │   ├── navbar.html
│   │   ├── dashboard.html
│   │   ├── subscriptions.html
│   │   ├── budgeting.html
│   │   └── spending-reports.html
├── documents/            # Project proposals, architecture docs, etc.
├── presentation slides/
├── reports/              # Documentation files
└── README.md             # Project overview
```

---

3. Build Instructions

- **Install Dependencies:**
  Navigate to the backend directory and run:
  ```
  npm install
  ```
- **Starting the Development Server:**
  For development, run:
  ```
  node index.js
  ```
  The application will be accessible at the localhost link in the console.

---

4. Setup and Deployment Instructions

- **Prerequisites:**

  - Node.js (version 14.x or later)
  - npm (bundled with Node.js)
  - MySQL (version 5.7 or later)
  - Git (Bash preferred)
- **Environment Setup:**

  - Clone the repository and change into the project directory.
  - Navigate to the backend directory:
    - Either using:
      ```
      cd personal\ finance\ analyzer/backend/
      ```
    - Or step-by-step:
      ```
      cd 'personal finance analyzer'
      cd backend
      ```
  - Install Node.js dependencies:
    ```
    npm install
    ```

    - **MySQL Database Setup:**
        - Create a new database in MySQL.
        - Run the SQL scripts (located in backend/src/databasemodels/schema.sql) to create the necessary tables.
    - **Plaid Integration:**
        - Create a Plaid account and obtain your Client ID and Secret.
    - **Configure Environment Variables:**

Create a .env file in the root directory with the following variables:

```
DB_HOST=<your-database-host>
DB_USER=<your-database-username>
DB_PASS=<your-database-password>
DB_NAME=<your-database-name>
PLAID_CLIENT_ID=<your-plaid-client-id>
PLAID_SECRET=<your-plaid-secret>
```

- **Running the Application:**

    - In the backend directory, start the server:
      node server.js
    - Access the application at the localhost link it logs in the console..

---

5. Testing and Continuous Integration

- **Testing Setup:**
    - **Back-end Tests:**
        - Navigate to the backend/tests/backend directory.
        - Run tests using Mocha & Chai:
          npm test
        - Example test file (transactions.test.js) verifies API endpoints.
    - **Front-end Tests:**
        - Located in backend/tests/frontend.
        - Use Jest and React Testing Library for testing components.
        - Run with:
          npm test
- **Continuous Integration (CI):**
    - **CI Service:** GitHub Actions is used to run CI/CD pipelines.
    - **CI Configuration:**
      A sample workflow file (.github/workflows/ci.yml) includes steps such as:
        - Checking out the repository.
        - Setting up Node.js.

- Installing dependencies.
- Running tests.

Code:

```
name: CI Pipeline
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
jobs:
  build:
    runs-on: ubuntu-latest
    services:
      mysql:
        image: mysql:5.7
        env:
          MYSQL_ROOT_PASSWORD: root
          MYSQL_DATABASE: testdb
          MYSQL_USER: testuser
          MYSQL_PASSWORD: testpass
        ports:
          - 3306:3306
        options: >-
          --health-cmd="mysqladmin ping --silent"
          --health-interval=10s
          --health-timeout=5s
          --health-retries=3
    env:
      DB_HOST: 127.0.0.1
      DB_USER: testuser
      DB_PASSWORD: testpass
      DB_NAME: testdb
      PLAID_CLIENT_ID: ${{ secrets.PLAID_CLIENT_ID }}
      PLAID_SECRET: ${{ secrets.PLAID_SECRET }}
      PLAID_ENV: sandbox
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
      - name: Set up Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'
      - name: Install dependencies
        working-directory: 'personal finance analyzer/backend'
        run: npm install
      - name: Run backend tests
        working-directory: 'personal finance analyzer/backend'
        run: npm run test:backend
      - name: Run frontend tests
        working-directory: 'personal finance analyzer/backend'
        run: npm run test:frontend
```

- **CI Triggers:**
    - On pushes and pull requests to the main branch.

---

6. Release Packaging

This release package for the source distribution includes:

- The complete source code with build scripts and environment configuration.
- Detailed release notes (this document) explaining the build process and development steps.
- A test package containing:
    - Unit tests for key components.
    - Integration/system tests covering essential use cases.
    - Instructions for running tests and viewing latest test results.

---

7. Assumptions / Toolset Requirements

Git Bash is assumed to be installed for a developer. Npm also needs to be installed in order to download dependencies and run tests. Node.js is required to start the server on your personal computer.

---

8. Issue Tracking and Change Logs

Github issues is available to be seen by going to the repository on github and clicking the Issues tab at the top. You can look through these, and sort through them by time to see history. We only began to use this recently, however, so there is not much history currently.