

# LANTERN

Presented by Aiden Gonzalez, Shibi  
Balamurugan, Tosin Ogunjobi



# The Lantern Team



**Shibi Balamurugan**

Senior  
Computer Engineering

**Backend and Frontend**



**Aiden Gonzalez**

Senior  
Computer Engineering

**API and Backend**



**Tosin Ogunjobi**

Senior  
Computer Engineering

**Frontend and UI**



**What is  
Lantern?**

# Overview

- A free, open source accounting and financial planning web application designed for small businesses
- **Why ?** According to US Bank, **82%** of small businesses fail due to poor cash flow management and organization
- Lantern aims to give a clear insight of financials for businesses owners with provided features and analysis
- In the future, we hope for small businesses to utilize our service and collaborate with other developers to further advance the application



## **Project Goal**

Create an interactive platform to allow users to view their financials through dashboards and reports and performs in real-time.

# S.M.A.R.T. Requirements

**Plaid Connectivity:** Utilizes Plaid, reliable and secure fintech platform, to allow for seamless integration into user's financial life

100% → 100%

**Insights:** Widgets, tools, and dashboard components to give user instant insights into their organization's financial well-being

90% → 100%

**Reporting:** Accurate financial reporting using user's current financial data

10% → 100%

**Dashboards:** Intuitive, customizable dashboards with accurate at-a-glance statistics

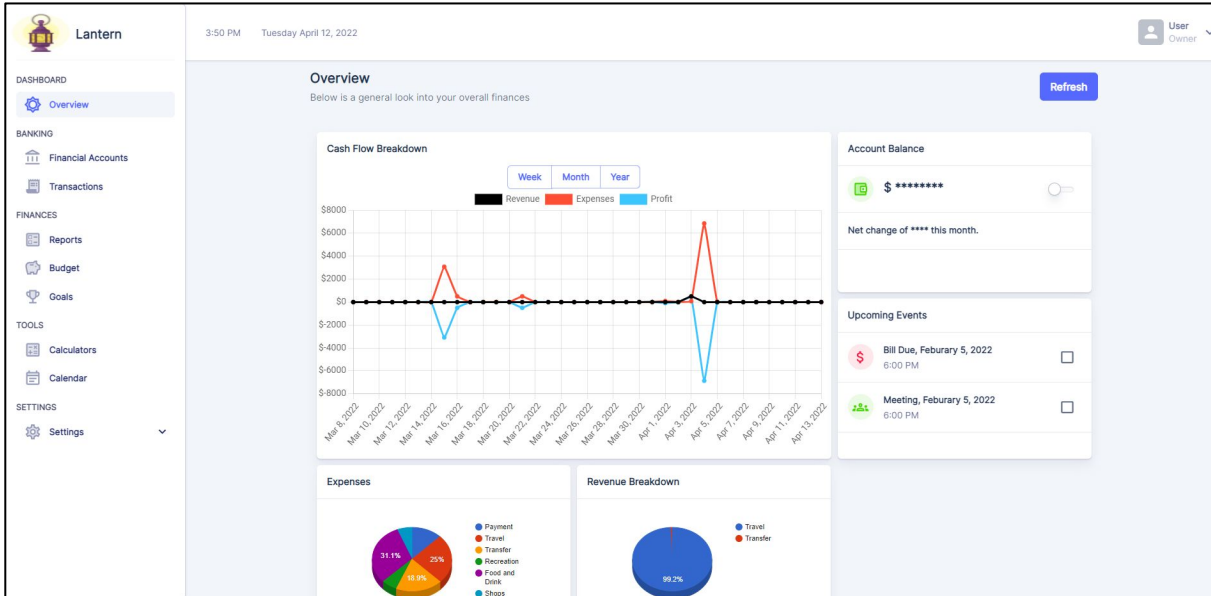
50% → 100%

**Performance:** Responsive interface and robust backend allow for seamless interaction and a highly scalable number of concurrent users

50% → 100%

# Features

## Dashboard -



- Total Account Balance
- Widgets -
  - Expenses Breakdown
  - Revenue Breakdown
  - Cash Flow Breakdown
  - Upcoming Events
- Draggable Components

# Financial Accounts and Transactions -

## Financial Accounts

Below are the bank accounts that you have linked with Lantern

+ Add Account

### Accounts

BANK NAME	ACCOUNT NAME	BALANCE	LATEST UPDATE	UPDATE
Fifth Third Bank	Plaid Checking	\$110.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid Saving	\$210.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid CD	\$1,000.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid Credit Card	\$410.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid Money Market	\$43,200.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid IRA	\$320.76	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid 401k	\$23,631.98	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid Student Loan	\$65,262.00	4/26/2022, 10:42 AM	
Fifth Third Bank	Plaid Mortgage	\$56,302.06	4/26/2022, 10:42 AM	

## Transactions

Below are your recent transactions from all connected accounts

### Filters

Start Date

to

End Date

Search

Category

All

DATE	TRANSACTION NAME	SOURCE	AMOUNT	CATEGORY
Wednesday, Apr 20 2022	United Airlines	*****jHlglL59	-\$500.00	Travel
Monday, Apr 18 2022	Uber 072515 SF**POOL**	****BCx5Xy3Q	-\$6.33	Travel
Friday, Apr 15 2022	Tectra Inc	*****jHlglL59	-\$500.00	Food and Drink
Thursday, Apr 14 2022	AUTOMATIC PAYMENT - THANK	*****jHlglL59	-\$2,078.50	Payment
Thursday, Apr 14 2022	KFC	*****jHlglL59	-\$500.00	Food and Drink

Rows per page: 5

1-5 of 386

< >

# Financial Reports -

## Expense Breakdown

Day Week Year

CATEGORY	PERCENTAGE OF TOTAL EXPENSES	NUMBER OF EXPENSES
Food and Drink	35.50%	120
Travel	21.89%	74
Payment	14.20%	48
Transfer	14.20%	48
Shops	7.10%	24
Recreation	7.10%	24

## Revenue Breakdown

Day Week Year

CATEGORY	PERCENTAGE OF TOTAL REVENUES	NUMBER OF REVENUES
Travel	50.00%	24
Transfer	50.00%	24

### Detailed Report

Total Transactions over Time Span: 48

Highest Categorized Revenue: Travel

Least Categorized Revenue: Transfer

Largest Transaction:

Name: United Airlines

Date: Wednesday May 13, 2020

Amount: -\$500.00

## Tools -

### Calculators

11:03 AM Tuesday April 26, 2022

#### Calculators

Below are some helpful calculators that will aid you in financial estimation and growth

##### Compound Annual Growth Rate

Growth rate (CAGR)	%
Number of periods	
Initial value	\$
Final value	\$
Difference	\$
Total growth	%

[Simple mode](#) [Reset defaults](#) [🔄](#) [Share result](#)



omni<sup>®</sup> CALCULATOR

##### Loan Payment Calculator

##### Return On Investment

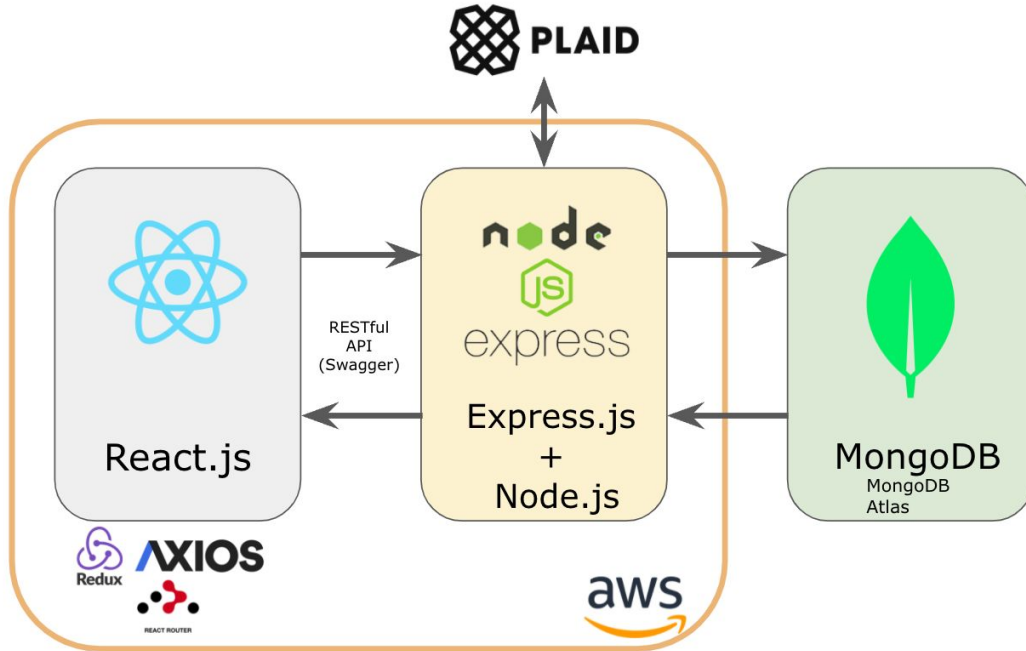
### Calendar

#### Upcoming Events

-  **Bill Due, February 5, 2022** 6:00 PM
-  **Meeting, February 5, 2022** 6:00 PM

# Software Architecture

## MERN Stack



- MERN
  - Frontend - React
  - Backend - Express + Node
  - Database - MongoDB
- MongoDB Atlas - Managed cloud NoSQL storage
- AWS hosting + EC2
- Plaid - Access financial data
- 100% JavaScript + TypeScript

# Open Source Development



# CI/CD

- **“Build” system:**
  - Node Package Manager (npm) used for managing dependencies for both the frontend and backend applications
  - Test and build commands used to automate common but important tasks
- **Continuous Integration:**
  - Used GitHub Actions workflows to automatically measure and enforce S.M.A.R.T. goals
- **Continuous Deployment:**
  - One-click deployment pipeline leveraging AWS CodeDeploy and PM2 to deploy and manage a hosted Lantern instance

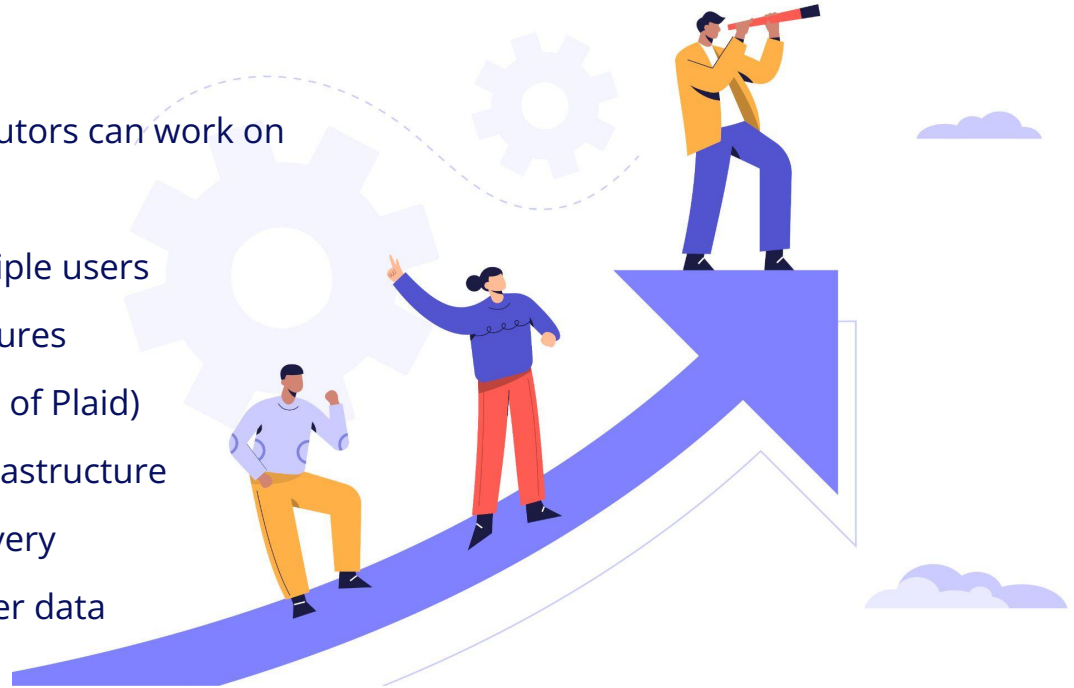




# Product Vision

# Future of Lantern

- Beta Release v1.0.0
- Making issues that outside contributors can work on
  - Detailed user profiles
  - Business accounts with multiple users
  - More robust accounting features
  - Custom data import (outside of Plaid)
- Fully deploy product onto AWS infrastructure
- Alternate methods of product delivery
- Upgrading Plaid Credentials for user data



# Skills Learned

- Technical
  - MERN Stack Experience (Mongo - Express - React.js - Node)
  - Designing and Writing Unit Tests and performing integration/end-to-end tests
  - Git/Github (50 PRs)
  - CI/CD
- Non-Technical
  - Market/Competitive Analysis
  - Long-term Project Management (94 completed issues)
  - Communicating Feature Requirements

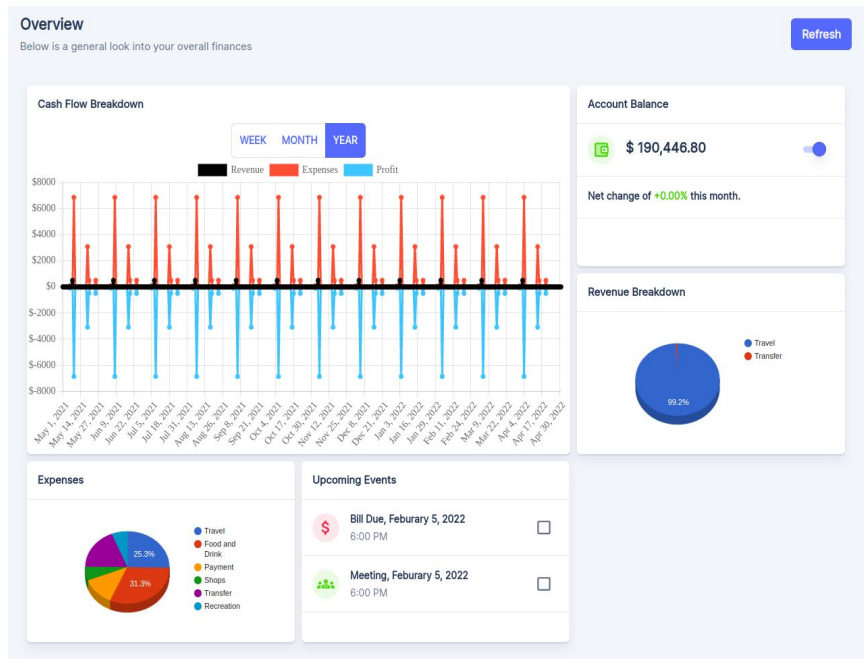




# Technical Challenges and Solutions

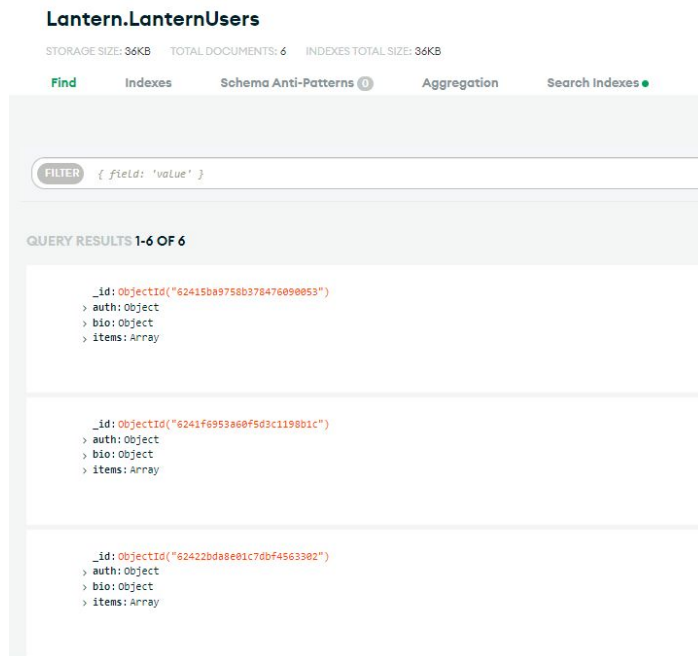
# Performance Issues

- Fetching a user's transactions was very slow, and was performed every time the user navigated to a different page or refreshed the current page
- Created a solution to pull in transactions in smaller batches, so that the UI can begin render useful information after just the first batch
- Redux saved transaction data to be shared with other components
- Automatic refresh periodically, manual refresh button available if desired



# Database Testing

- Unit testing with MongoDB required us to create and teardown MongoDB collections
- Used temporary in-memory database for testing
  - Allowed us to make our tests much cheaper (less time and less Atlas cost)
  - Also could avoid any potential network issues trying to connect to Atlas
- For more transparency, we added a log of the in-memory database to be saved to the local disk for post-test inspection and analysis, as-needed



**Lantern.LanternUsers**  
STORAGE SIZE: 36KB TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

**FILTER** { field: 'value' }

**QUERY RESULTS 1-6 OF 6**

```
  _id: ObjectId("62415ba9758b378476090053")
  > auth: Object
  > bio: Object
  > items: Array

  _id: ObjectId("6241f6953a60f5d3c1198b1c")
  > auth: Object
  > bio: Object
  > items: Array

  _id: ObjectId("62422bda8e01c70bf4563302")
  > auth: Object
  > bio: Object
  > items: Array
```

# Standardizing Client-Server Interaction

- Working on frontend and backend in parallel
  - Needed a clearly defined API specification
  - As long as the interface was clearly defined, both sides could implement and test new features independently
- Used open-source API design tool “apicurio”
  - Easy graphical interface to define OpenAPI specification
  - Automatically generate documentation, code snippets, and tests
- Hosted site enabled collaboration, so that the spec could be an evolving document throughout development

The image shows a screenshot of an API documentation tool. It displays two endpoints with their descriptions and response samples.

**GET /api/accounts**  
Gets a list of all `.Account` entities for user owning provided token

AUTHORIZATIONS: clientToken

Responses

- > 200 Successful response - returns an array of `.Account` entities.
- > 400 No valid Plaid credentials in account! Cannot complete request.
- 401 No ClientAuthToken provided
- 403 Invalid ClientAuthToken!
- > 500 Database/Plaid error

**Response samples**

200 400 500

Content type: application/json

```
{
  - {
    "name": "string",
    "description": "string",
    "id": "string",
    "balance": 0,
    + "institution": { - }
  }
}
```

Copy Expand all Collapse all

---

**GET /api/link**  
The first step of linking a bank account. Server will request a `link_token` from Plaid `/link/token/create` endpoint, and return the temporary token to the client.

AUTHORIZATIONS: clientToken

Responses

- > 200 Successfully created link token
- 401 No ClientAuthToken provided
- 403 ClientAuthToken invalid
- > 500 Database/Plaid error

**Response samples**

200 500

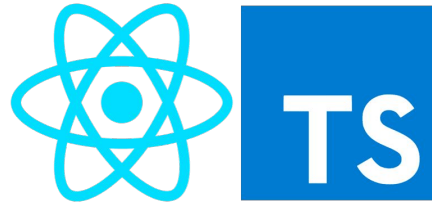
Content type: application/json

```
{
  "token": "string"
}
```

Copy Expand all Collapse all

# Converting Ideas to Code

- After brainstorming new ideas, we had to find ways to implement them using Typescript and React
- Front-End Development very different from ECE undergraduate courses
- “What Should We Do Next?”
- Using online resources and asking questions was a very productive to come up with solutions





**DEMONSTRATION**

**Questions ?**



**THANK YOU  
FOR  
LISTENING**