

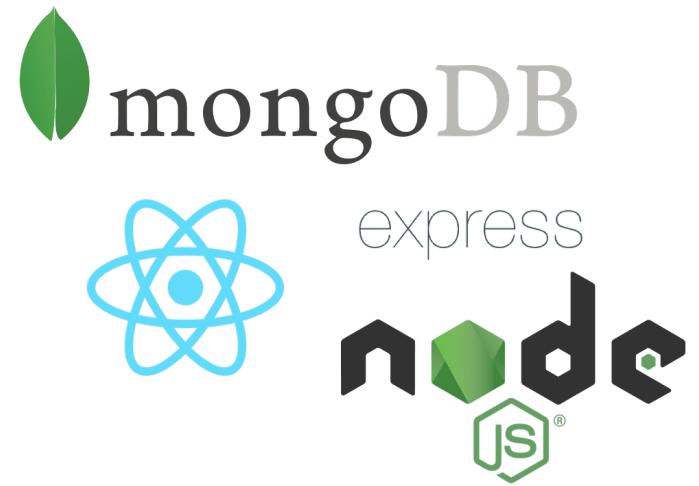


WATCH YOUR WALLET

By: Adriana & Hitanshi

Our Project:

- For our final project, we are creating a web application called “Watch Your Wallet”
- It is a budgeting app designed specifically for students.
- To create this project we are using the MERN stack, which includes:
 - MongoDB, Express.Js, React.JS, and Node.JS
 - MongoDB: used to store our users transaction information
 - Express.JS: used for the backend of our application
 - React.JS used for the frontend of our application
 - Node.JS – used to launch both the frontend and backend of our web app using JavaScript



Target Market Selection

In order to come up with an idea, we interviewed a few of our friends about their current struggles

Most of them mentioned having problems with managing their finances

A lot of them stated that they used their credit card, debit cards, cash, and sometimes paid for their friends or had their friends pay on their behalf

Since there were many options, they had difficulty keeping a track of their finances

Watch Your Wallet

- Based on survey we decided to create this finance app called “Watch your Wallet” – its essentially a web app that lets users manually **enter all of their transactions.**
- We let our users enter a name for their transaction that's **easy for them remember.**
- They are also allowed to **modify and delete previous transactions** if needed.

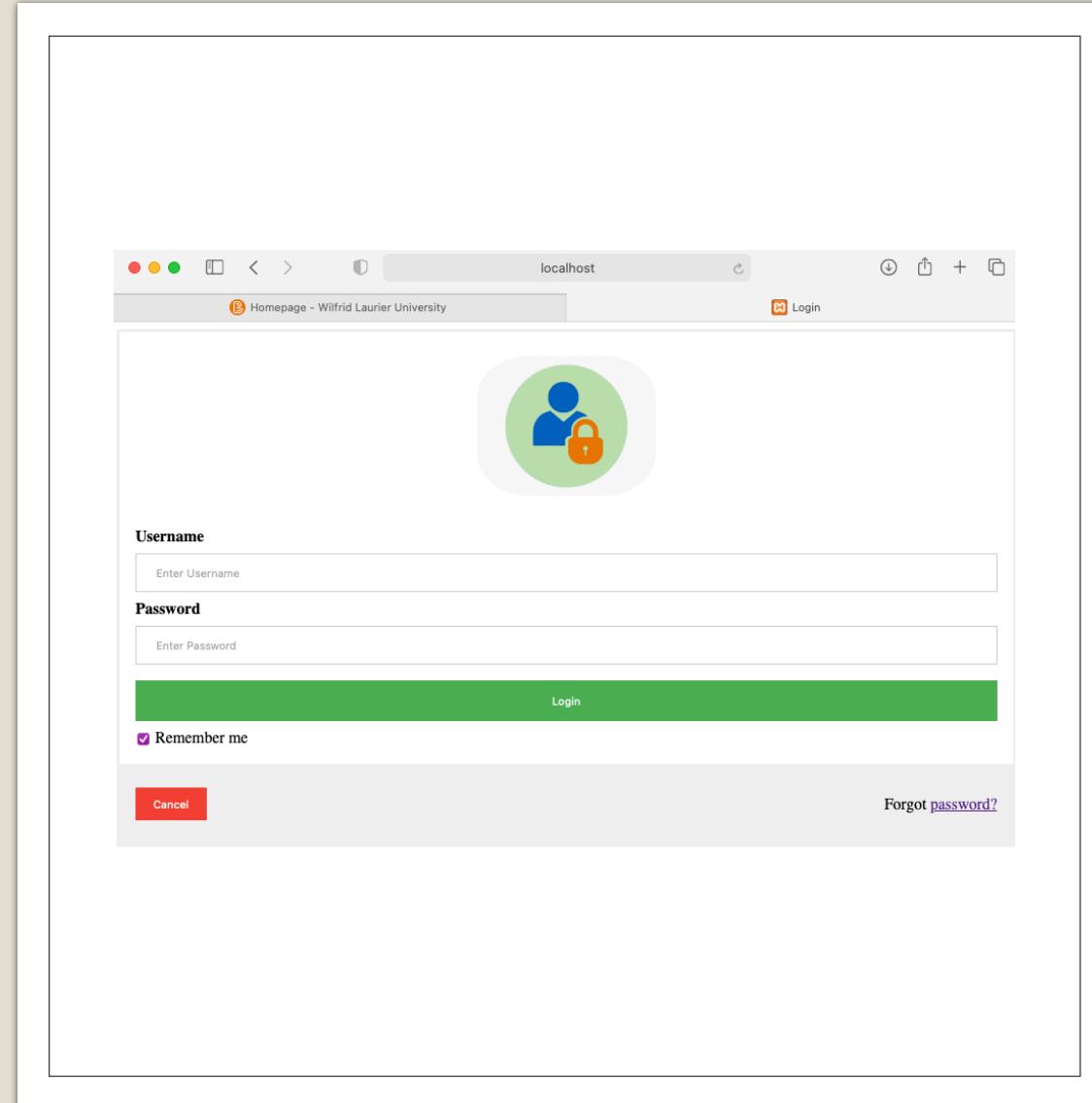


Watch Your Wallet

Functionality

- Login
- Homepage
- Transactions
- Addition of Transactions
- Removal of Transactions
- Updating Balance
- Updating Specific Accounts:
 - Income & Expense

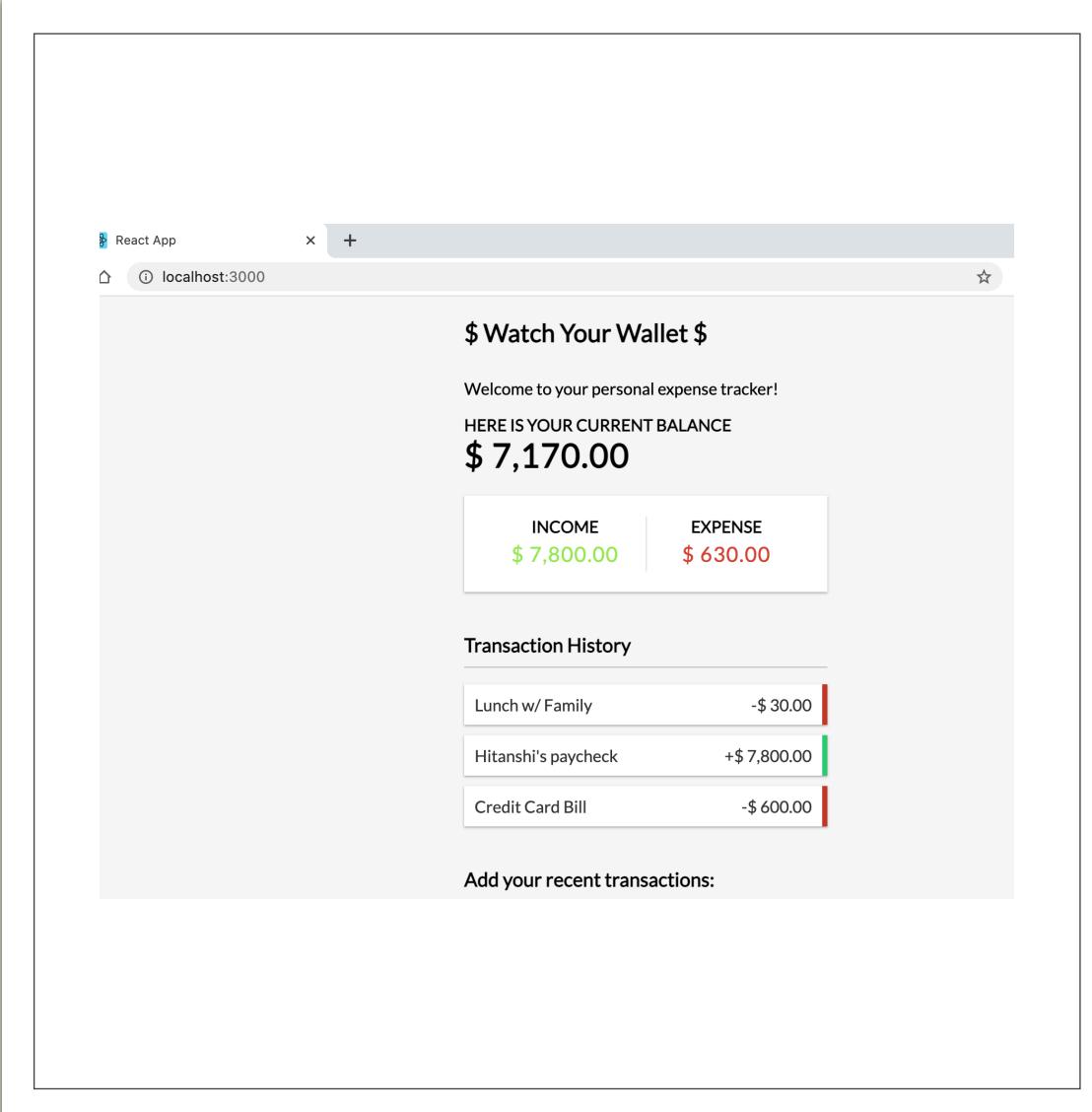
Login



A screenshot of a web browser window titled "localhost". The address bar shows "Homepage - Wilfrid Laurier University". The main content area displays a login form. At the top is a green circular icon containing a blue user profile and an orange padlock. Below it are two input fields: "Username" (placeholder "Enter Username") and "Password" (placeholder "Enter Password"). A large green "Login" button is centered below the password field. To the left of the "Login" button is a checked checkbox labeled "Remember me". At the bottom left is a red "Cancel" button, and at the bottom right is a link "Forgot [password?](#)".

- Initially the user is expected to login to access their account.
- They have to enter their username and password and click login to be allowed to view their homepage.
- They can select “Remember me” to have their password saved and “Forgot password?” to reset their password.

Homepage



- Once the login, they are brought to our react app.
- Here they can visually see their current balance.
- They can see their income and expenses and their transaction history.

Transaction History

Lunch w/ Family	-\$ 30.00
Hitanshi's paycheck	+\$ 7,800.00
Credit Card Bill	-\$ 600.00

Add your recent transactions:

Transaction label/name

Amount
(negative - expense, positive - income)

Record Transaction

Transactions

- The transaction history keeps a record of all of their transactions.
- This information is stored in MongoDB.
- All of the money flowing our of the account (the negatives) have red on the history label and all of the money flowing into the account (the positives) have a green.

Adding Transactions

- We have made the UI fairly straight forward to avoid any confusion and make the entire experience of using our web app easy & fun.
- In order to add a transaction:
 - The user first enters the transaction label/name
 - Enters the amount manually (or they can use the arrows)
 - Then clicks on record transaction

Add your recent transactions:

Transaction label/name

Amount
(negative - expense, positive - income)

Record Transaction

Amount
(negative - expense, positive - income)

Record Transaction

Welcome to your personal expense tracker!

HERE IS YOUR CURRENT BALANCE
\$ 7,070.00

INCOME \$ 7,800.00	EXPENSE \$ 730.00
-----------------------	----------------------

Transaction History

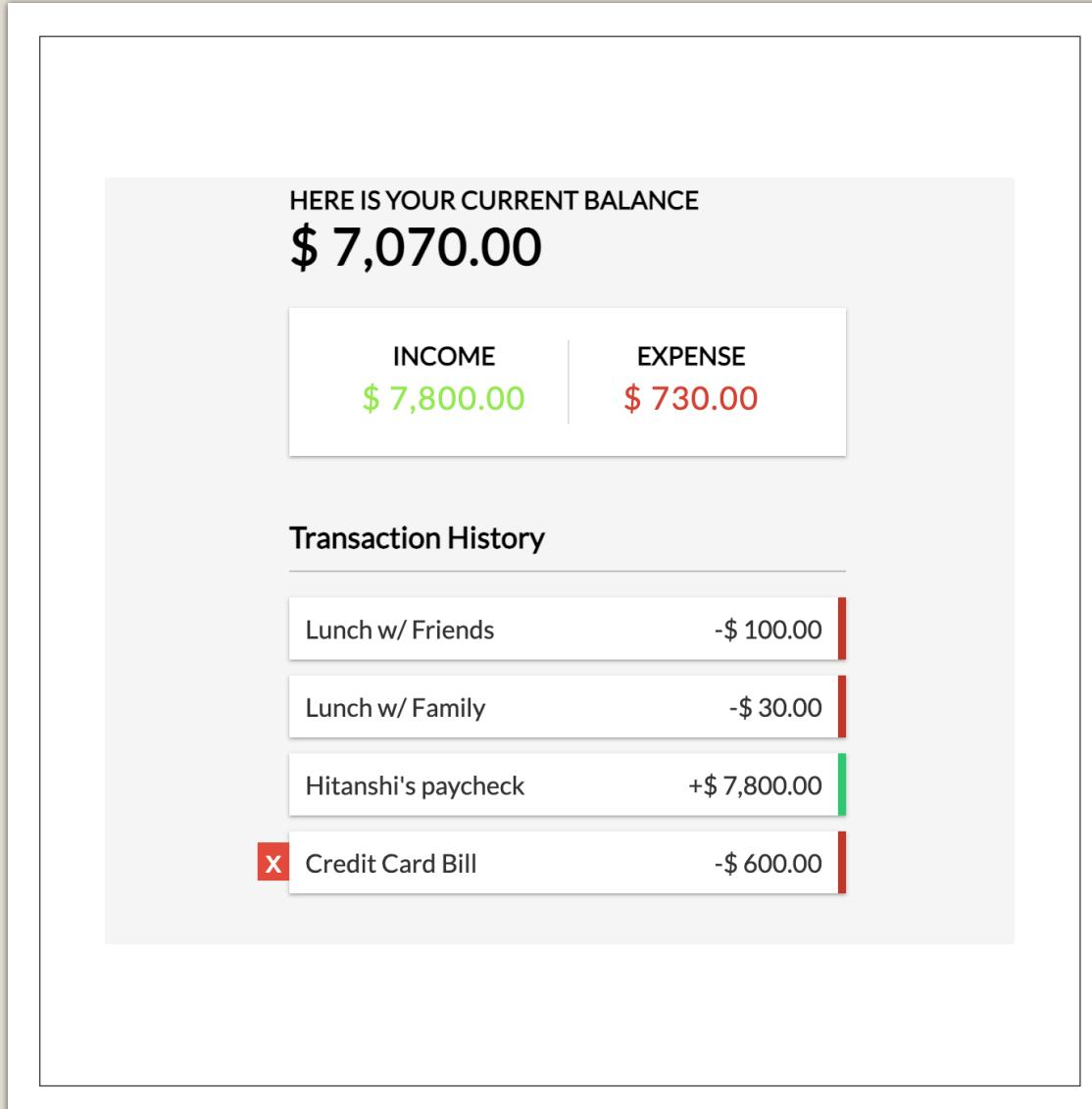
Lunch w/ Friends	-\$ 100.00
Lunch w/ Family	-\$ 30.00
Hitanshi's paycheck	+\$ 7,800.00
Credit Card Bill	-\$ 600.00

Update Balance

- Here we can see that the transaction from the last slide, “Lunch w/Friends” was added.
- \$100 was deducted from the current balance.
- And \$100 was added to the expense account.

Removing Transactions

- The user can easily remove a transaction and update the transaction history by browsing over the specific item and clicking the “x”



\$ Watch Your Wallet \$

Welcome to your personal expense tracker!

HERE IS YOUR CURRENT BALANCE

\$ 7,670.00

INCOME

\$ 7,800.00

EXPENSE

\$ 130.00

Transaction History

Lunch w/ Friends -\$ 100.00

Lunch w/ Family -\$ 30.00

Hitanshi's paycheck +\$ 7,800.00

Balance Update Pt. 2

- The app quickly updates the balance and expense section once a transaction has been removed.
- Here the Current Balance increases and the Expense account balance decreases since we deleted the “Credit Card Bill” transaction.

MongoDB

- We are using MongoDB to store the user's information in a database, specifically we are using clusters in MongoDB
 - The clusters allow us to ensure that each node does not handle all of the data, so we can separate data along all of the nodes if needed.
- Refer to the pictures in the next two slides, for further explanations/visuals.

MongoDB Cluster

cloud.mongodb.com/v2/606faab80eb8f82f81e9d5d0#clusters/detail/Cluster0

Wilfrid Laurier Univ... Access Manager Support Billing All Clusters Adriana

wyw Atlas Realm Charts

DATA STORAGE WILFRID LAURIER UNIVERSITY > WYW > CLUSTERS

Cluster0 VERSION 4.4.4 REGION AWS N. Virginia (us-east-1) CLUSTER TIER M0 Sandbox (General)

Clusters Triggers Data Lake

SECURITY Database Access Network Access Advanced

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

SANDBOX NODES REPLICA SET CONNECT CONFIGURATION ...

REGION N. Virginia (us-east-1)

- cluster0-shard-00-00.lvsq... SECONDARY
- cluster0-shard-00-01.lvsq... SECONDARY
- cluster0-shard-00-02.lvsq... PRIMARY

This is a Shared Tier Cluster
If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.
Upgrade

Operations R: 0 W: 0 0.04/s
Last 6 Hours

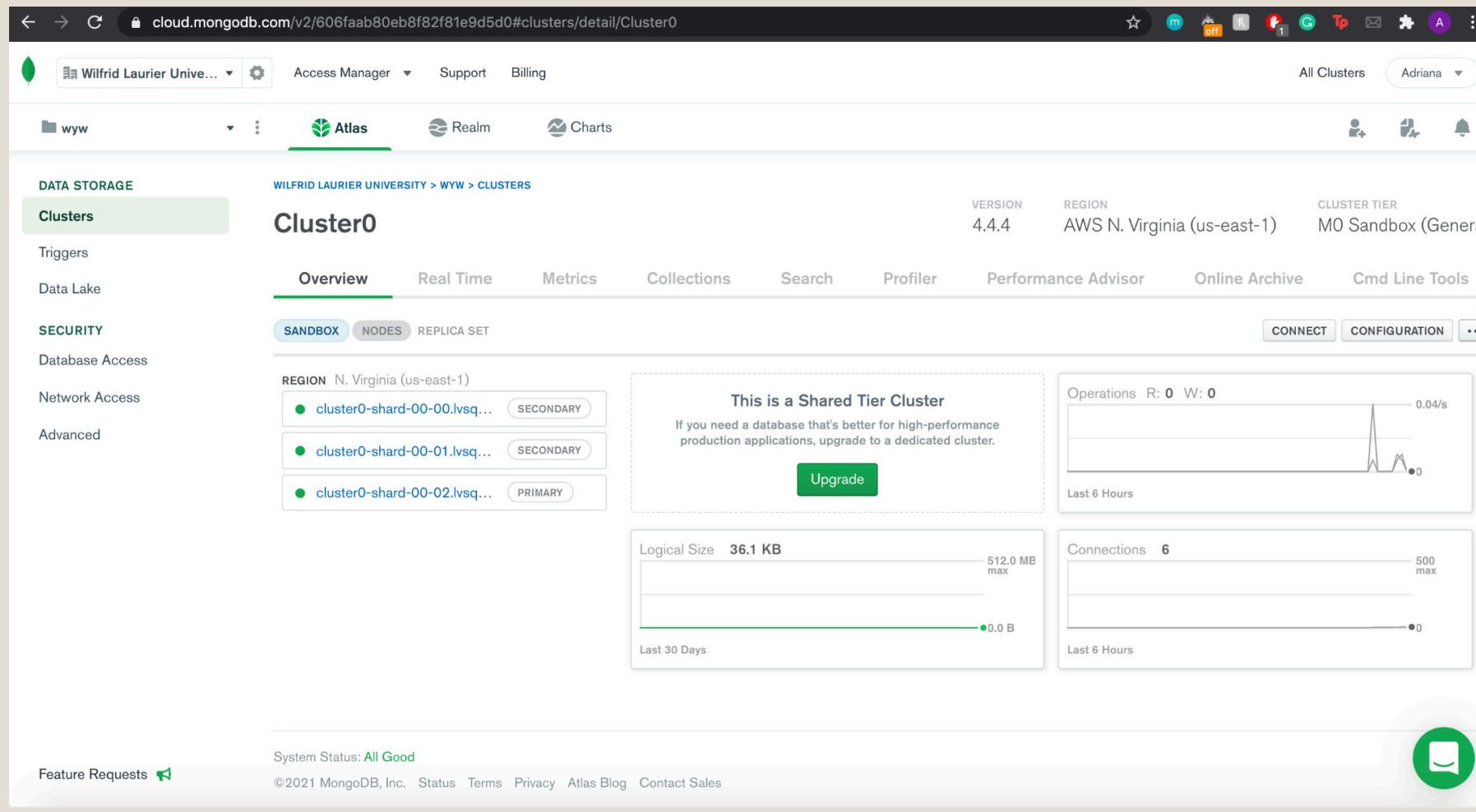
Logical Size 36.1 KB 512.0 MB max
0.0 B Last 30 Days

Connections 6 500 max
Last 6 Hours

System Status: All Good

Feature Requests 🔔

©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

A screenshot of the MongoDB Cloud interface showing the overview of a cluster named 'Cluster0'. The cluster is running version 4.4.4 in the AWS N. Virginia region with a M0 Sandbox tier. It consists of three shards: two secondary shards and one primary shard. The shards are labeled 'cluster0-shard-00-00.lvsq...', 'cluster0-shard-00-01.lvsq...', and 'cluster0-shard-00-02.lvsq...'. A callout box highlights that this is a 'Shared Tier Cluster' and suggests upgrading to a dedicated cluster for high-performance production applications. The 'Upgrade' button is prominently displayed. The page also shows real-time operations (R: 0, W: 0), a graph of operations over the last 6 hours, logical size (36.1 KB), and connection statistics (6 connections out of 500 max). The bottom of the screen displays the system status as 'All Good' and includes links for feature requests, status, terms, privacy, atlas blog, and contact sales.

The screenshot shows the MongoDB Atlas interface. In the top navigation bar, there are links for Manager, Support, and Billing. Below the navigation, there are tabs for Databases, Realm, and Charts. A sidebar on the left has a 'Create Database' button and sections for SPACES and tracker. The main area is titled 'expensetracker.transactions' and displays the following information: COLLECTION SIZE: 349B, TOTAL DOCUMENTS: 4, INDEXES TOTAL SIZE: 36KB. Below this, there are buttons for Find, Indexes, Schema Anti-Patterns (0), Aggregation, and Search. A 'FILTER' button with the value '{"filter": "example"}' is present. The 'QUERY RESULTS 1-4 OF 4' section lists four documents:

- `_id: ObjectId("607378429860ccab80c46d67")
text: "Hitanshi's paycheck"
amount: 7800
createdAt: 2021-04-11T22:29:22.630+00:00
__v: 0`
- `_id: ObjectId("6073784e9860ccab80c46d68")
text: "Shopping"
amount: -600
createdAt: 2021-04-11T22:29:34.211+00:00
__v: 0`
- `_id: ObjectId("607378599860ccab80c46d69")
text: "Credit Card Bill"
amount: -600
createdAt: 2021-04-11T22:29:45.983+00:00
__v: 0`
- `_id: ObjectId("607378699860ccab80c46d6a")
text: "Adriana's Rent"
amount: -500
createdAt: 2021-04-11T22:30:01.602+00:00
__v: 0`

MongoDB – Our Database

- The transaction information is being stored in our `expensetracker.transactions` database.
- Any transactions added by the user are added to the database and any transactions deleted by the user are removed from the database.

Thank you!

