PROBLEM STATEMENT:

Effective **budget management** is a common challenge for many. The core problems this project aims to address are:

- Unconscious Overspending: Many people overspend without a clear, real-time understanding of their financial habits.
- Inefficiency of Manual Budgeting: Traditional methods of manually tracking expenses are often time-consuming, tedious, and prone to human error.
- Difficulty in Future Planning: Accurately estimating future expenses is difficult, making long-term financial planning and savings a significant challenge.

PROPOSED SOLUTION

The **BudgetWise,** AI-Based Expense Forecasting Tool is a smart, data-driven application designed to help individuals and businesses manage their finances proactively. The tool tracks user expenses, automatically categorizes them, and leverages Artificial Intelligence (AI) and Machine Learning (ML) models to forecast future spending patterns. By providing predictive insights and actionable alerts, it empowers users to maintain their budgets, anticipate overspending, and make informed financial decisions.

BUDGETWISE

OBJECTIVES

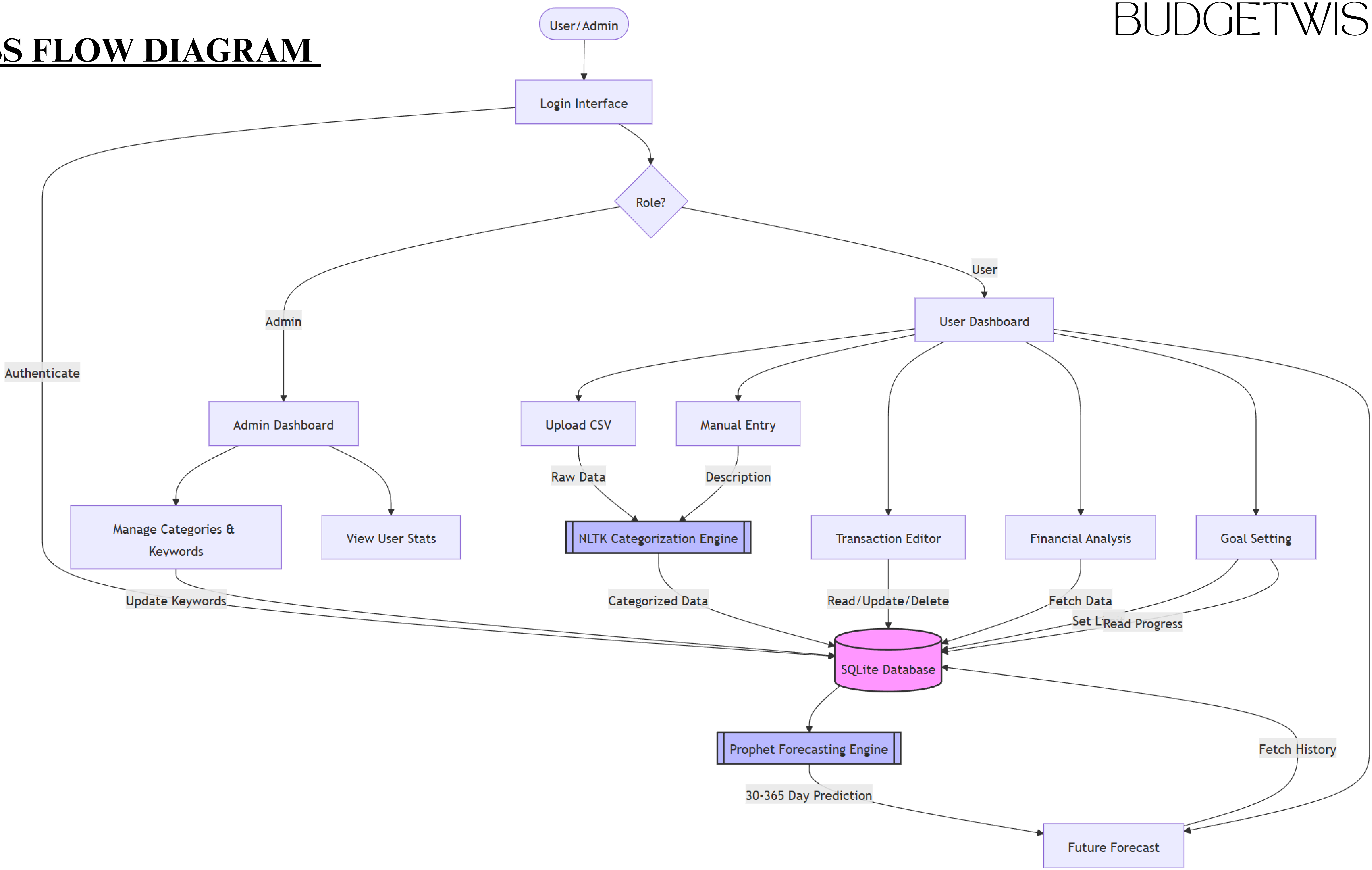
- **Automate Financial Hygiene**: To drastically reduce the time users spend cleaning and categorizing their bank statements.
- **Empower Proactive Planning**: To provide users with accurate forecasts, allowing them to adjust budgets before overspending occurs.
- **Ensure Data Integrity & Security**: To implement a robust backend (SQLite) that handles large datasets efficiently while maintaining strict separation between user data and administrative controls.
- **Actionable Goal Tracking**: To bridge the gap between planning and reality by enabling category-specific spending limits with real-time progress bars and alerts.

BUDGETWISE

LIST OF FEATURES OFFERED

- **Role-Based Authentication**: Secure login system with distinct dashboards for Users (personal finance) and Admin (system management).
- **NLP-Powered Categorization**: Automatically tags transactions with categories using NLTK tokenization and dynamic keyword matching.
- **Advanced Forecasting Engine**: Predicts future spending trends and seasonality up to 365 days ahead using Prophet model.
- **Interactive Data Management**: Features a robust CSV uploader for large datasets and a smart, paginated editor for managing transaction records.
- **Goal Settings**: Enables users to set monthly category budgets with real-time progress tracking and over-budget warnings.
- **Admin Control Panel**: Provides system-wide oversight and allows admins to instantly update categorization logic.

PROCESS FLOW DIAGRAM



TECHNOLOGIES USED IN THE SOLUTION

Python

Core Backend Logic

Streamlit

Interactive Web Interface

SQLite

Local Data Storage

Pandas

Data Analysis & Cleaning

NLTK

Automated Text Categorization

Prophet

Time-Series Forecasting

Matplotlib

Static Statistical Visuals

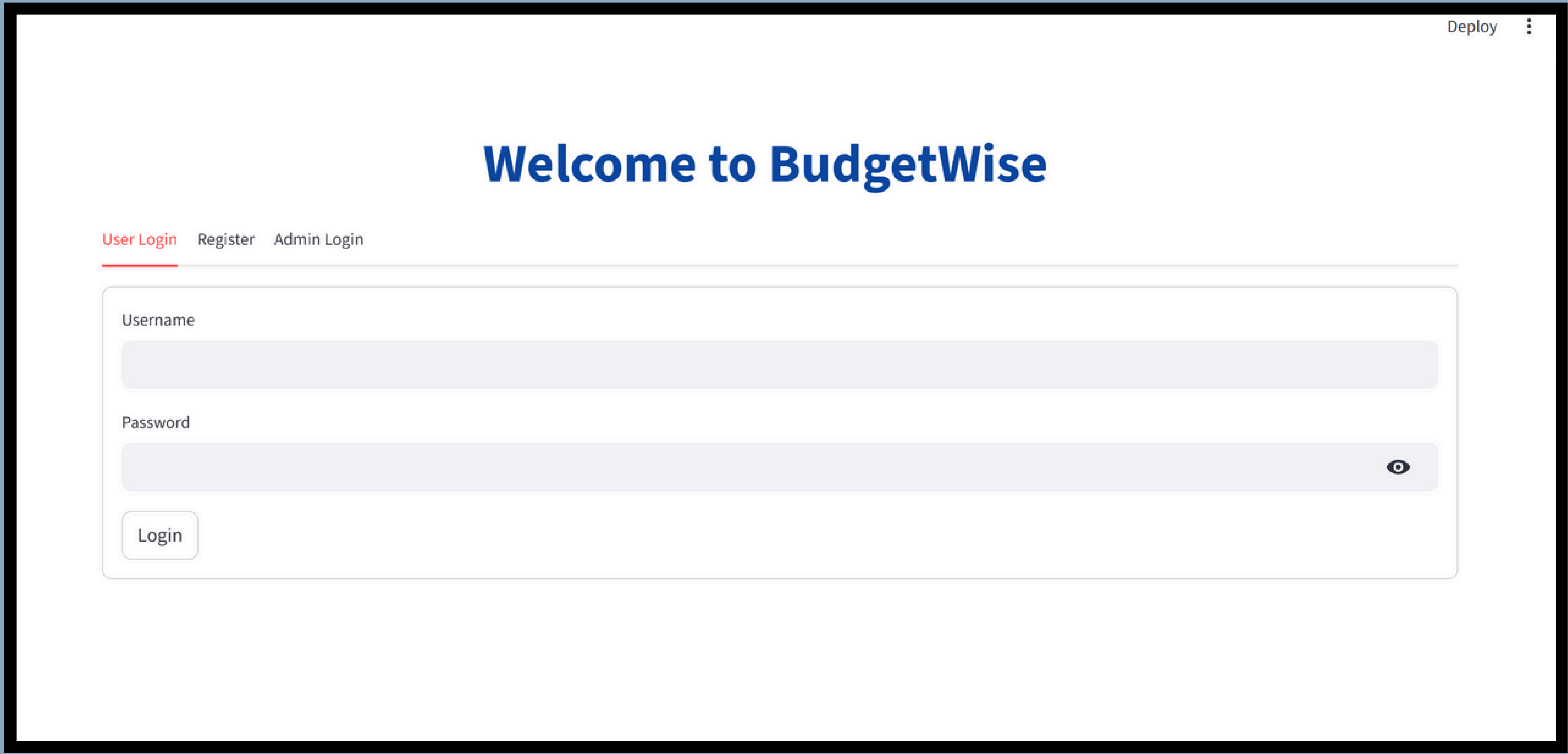
Altair

Interactive Forecast Charts

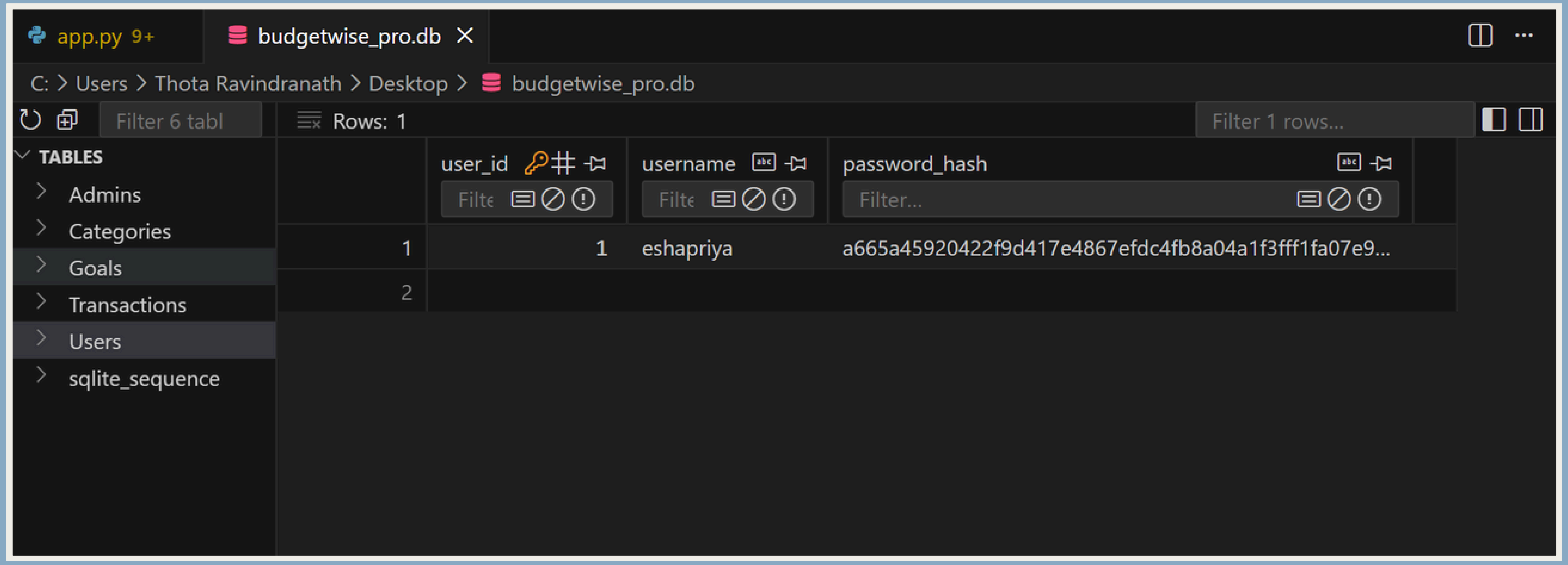
BUDGETWISE

MILESTONE 1:

- Established a secure Role-Based Authentication system using hashed passwords and SQLite storage.
- Developed a robust Manual Transaction Interface that validates inputs and instantly updates the local database.

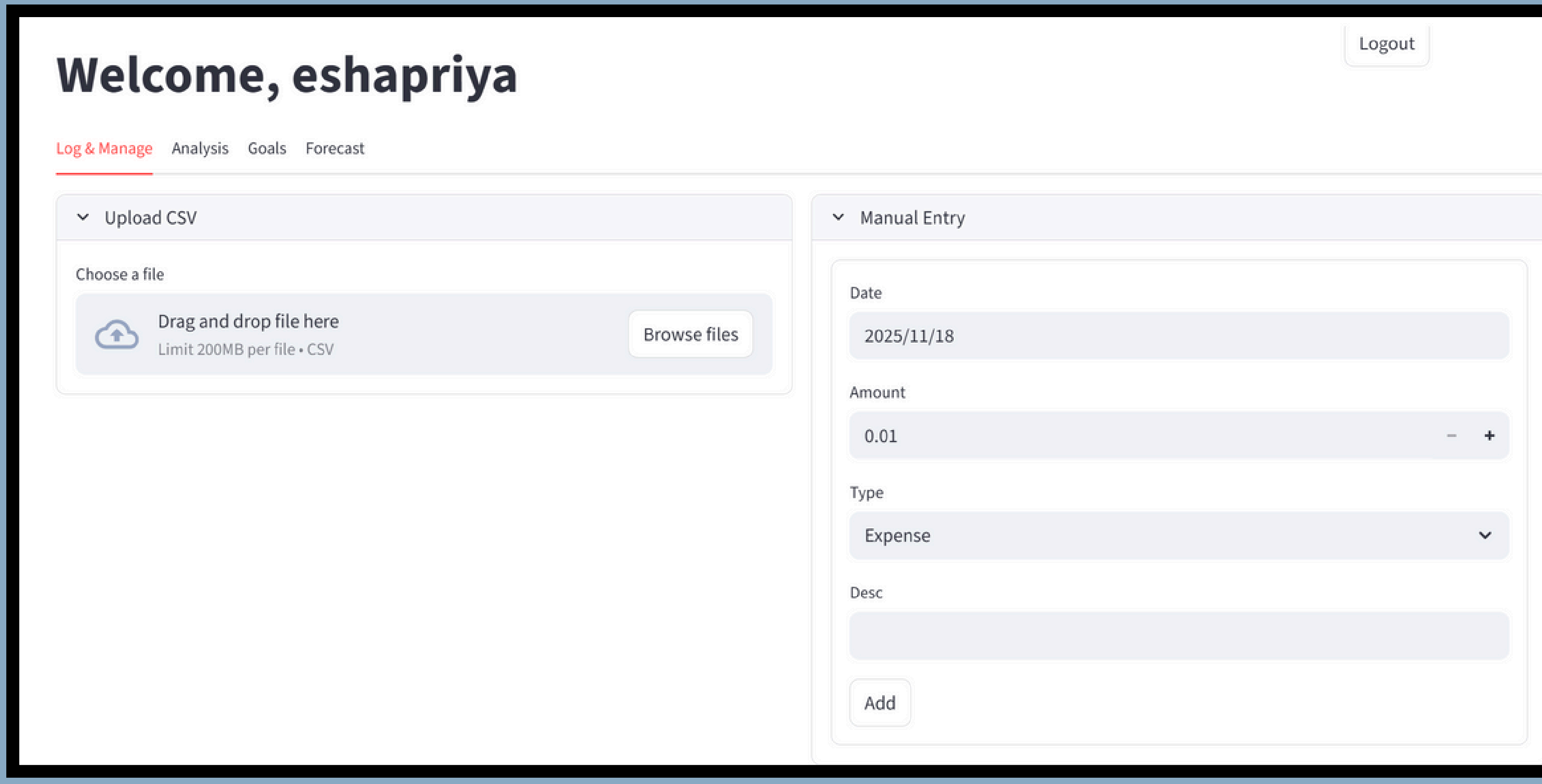


Welcome to BudgetWise



Rows: 1

	user_id	username	password_hash
1	1	eshapriya	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e9...
2			



Welcome, eshapriya

Date: 2025/11/18

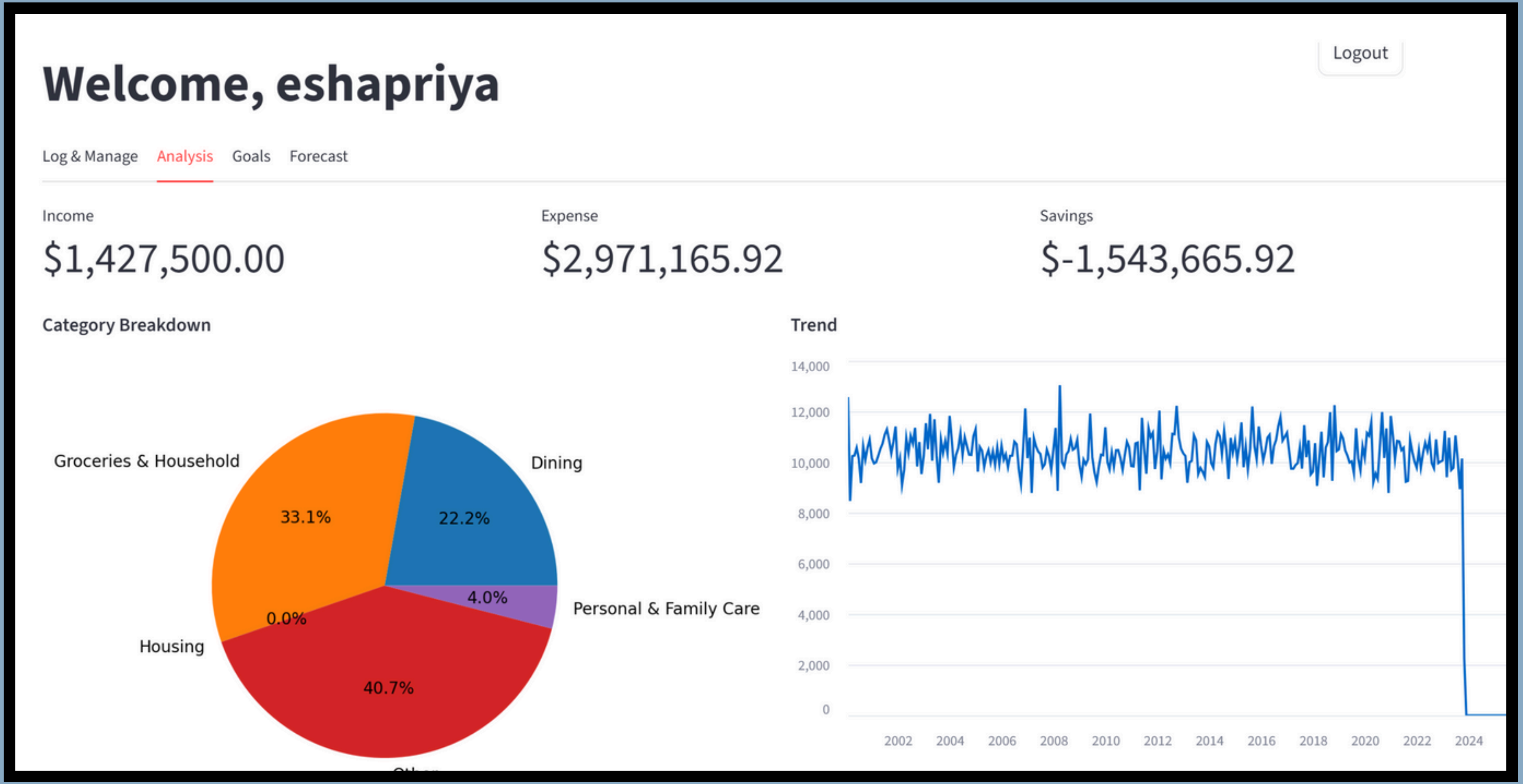
Amount: 0.01

Type: Expense

Desc:

MILESTONE 2:

- Integrated NLTK to build an NLP-powered engine that automatically categorizes transactions based on descriptions.
- Created a comprehensive Financial Dashboard featuring interactive charts for income vs. expense tracking and category breakdowns.



Welcome, eshapriya

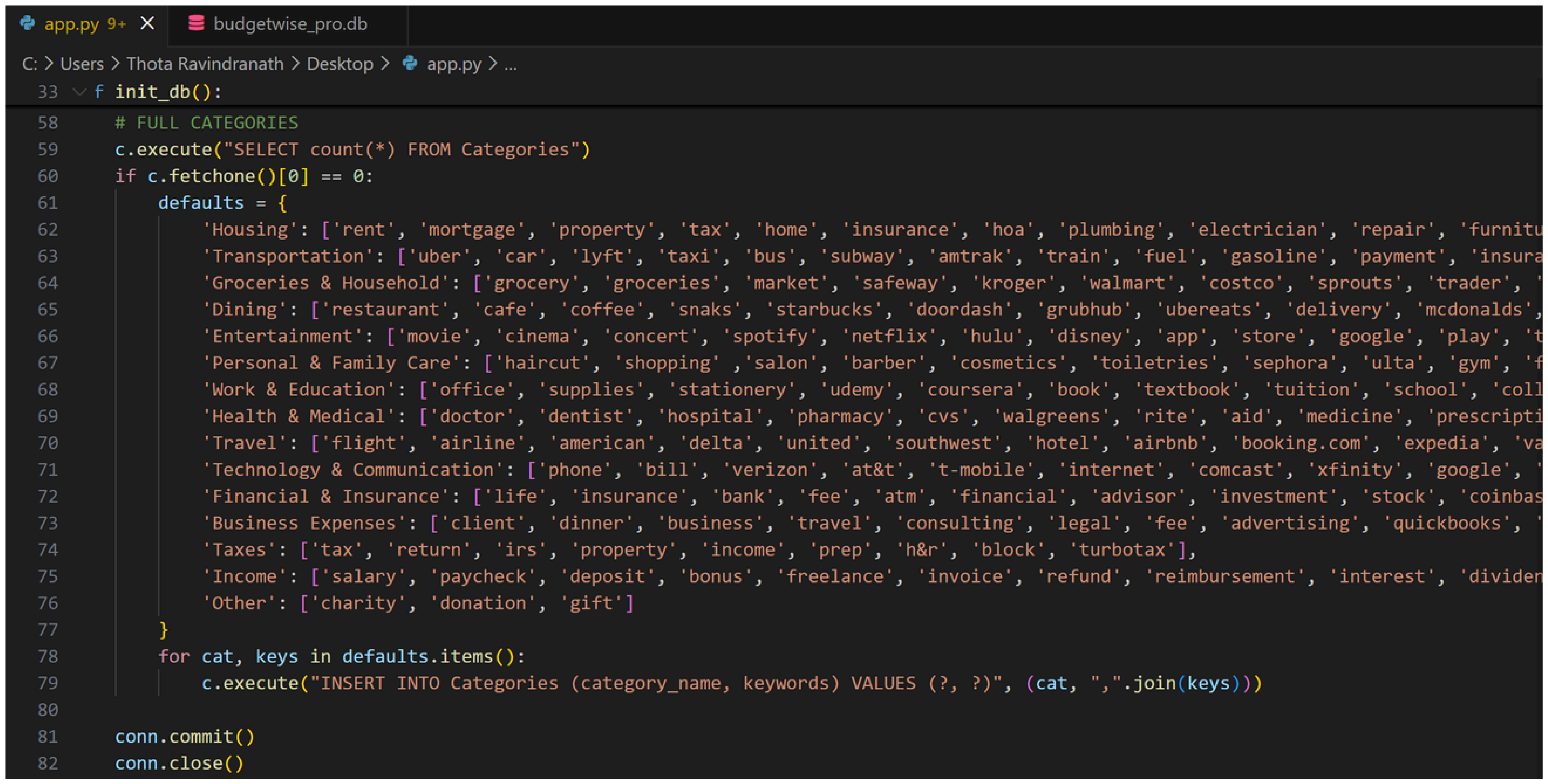
Income
$1,427,500.00

Expense
$2,971,165.92

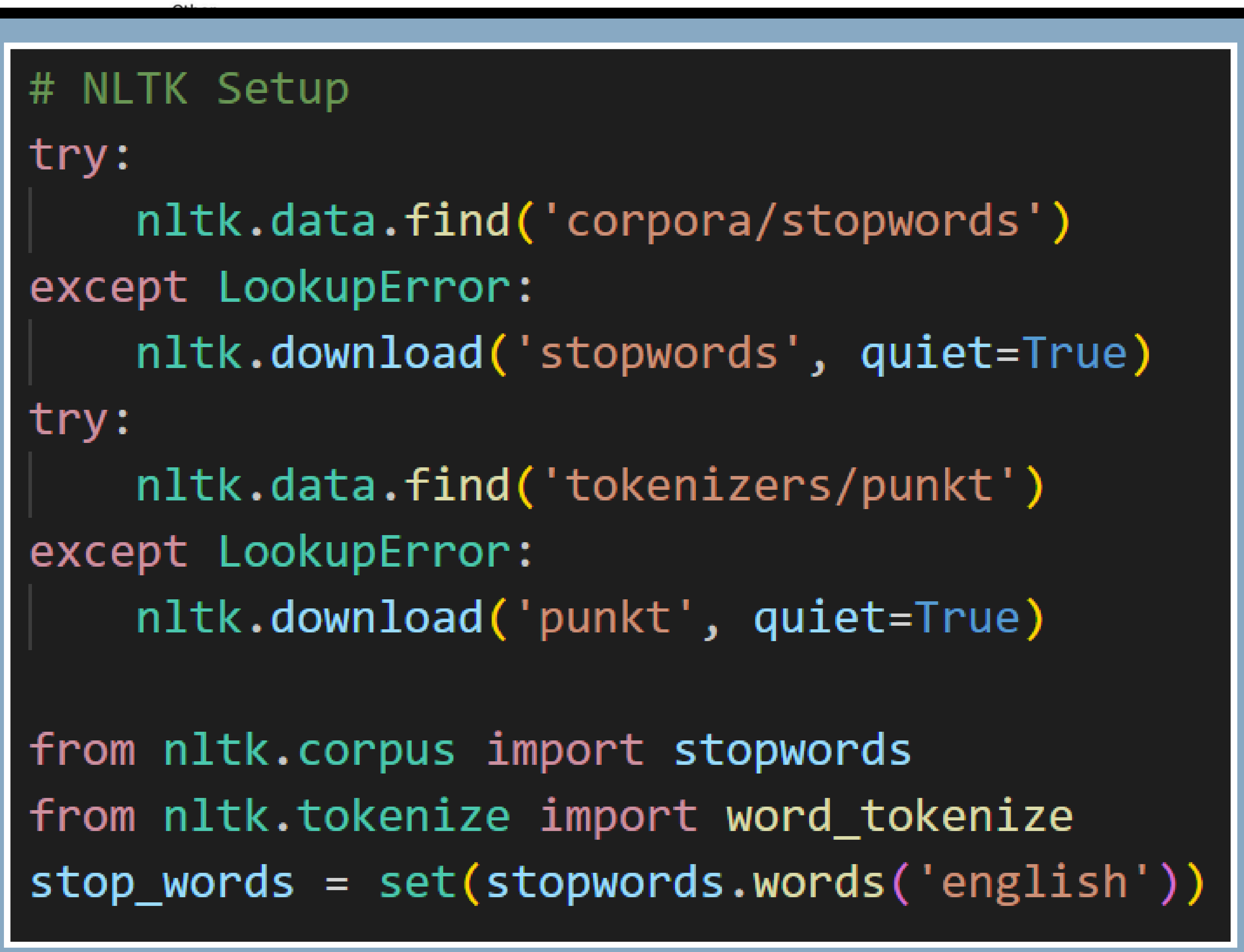
Savings
$-1,543,665.92

Category Breakdown

Trend



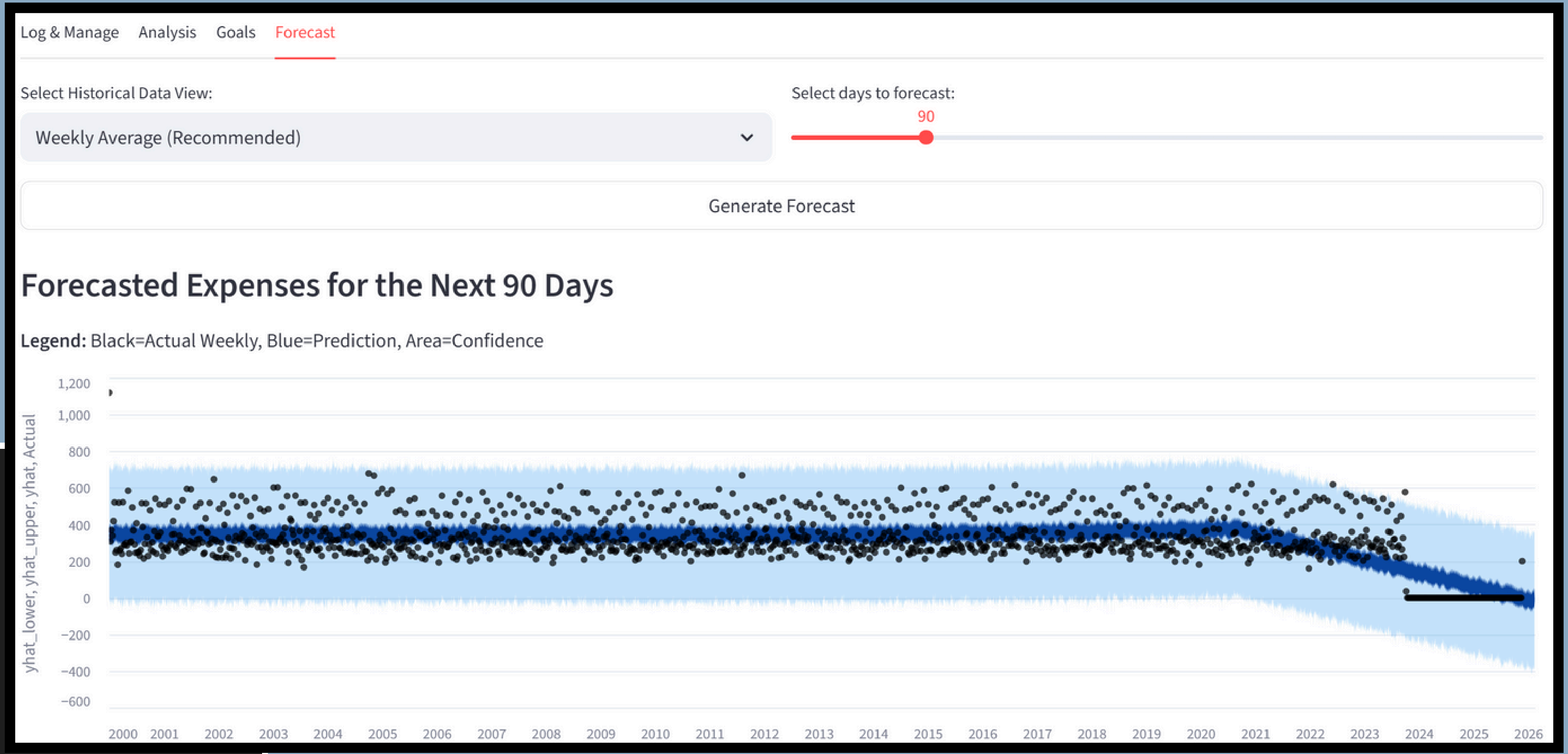
```python
f init_db():
    # FULL CATEGORIES
    c.execute("SELECT count(*) FROM Categories")
    if c.fetchone()[0] == 0:
        defaults = {
            'Housing': ['rent', 'mortgage', 'property', 'tax', 'home', 'insurance', 'hoa', 'plumbing', 'electrician', 'repair', 'furnitu
            'Transportation': ['uber', 'car', 'lyft', 'taxi', 'bus', 'subway', 'amtrak', 'train', 'fuel', 'gasoline', 'payment', 'insura
            'Groceries & Household': ['grocery', 'groceries', 'market', 'safeway', 'kroger', 'walmart', 'costco', 'sprouts', 'trader', '
            'Dining': ['restaurant', 'cafe', 'coffee', 'snaks', 'starbucks', 'doordash', 'grubhub', 'ubereats', 'delivery', 'mcdonalds',
            'Entertainment': ['movie', 'cinema', 'concert', 'spotify', 'netflix', 'hulu', 'disney', 'app', 'store', 'google', 'play', 't
            'Personal & Family Care': ['haircut', 'shopping' ,'salon', 'barber', 'cosmetics', 'toiletries', 'sephora', 'ulta', 'gym', 'f
            'Work & Education': ['office', 'supplies', 'stationery', 'udemy', 'coursera', 'book', 'textbook', 'tuition', 'school', 'coll
            'Health & Medical': ['doctor', 'dentist', 'hospital', 'pharmacy', 'cvs', 'walgreens', 'rite', 'aid', 'medicine', 'prescripti
            'Travel': ['flight', 'airline', 'american', 'delta', 'united', 'southwest', 'hotel', 'airbnb', 'booking.com', 'expedia', 'va
            'Technology & Communication': ['phone', 'bill', 'verizon', 'at&t', 't-mobile', 'internet', 'comcast', 'xfinity', 'google', '
            'Financial & Insurance': ['life', 'insurance', 'bank', 'fee', 'atm', 'financial', 'advisor', 'investment', 'stock', 'coinbas
            'Business Expenses': ['client', 'dinner', 'business', 'travel', 'consulting', 'legal', 'fee', 'advertising', 'quickbooks', '
            'Taxes': ['tax', 'return', 'irs', 'property', 'income', 'prep', 'h&r', 'block', 'turbotax'],
            'Income': ['salary', 'paycheck', 'deposit', 'bonus', 'freelance', 'invoice', 'refund', 'reimbursement', 'interest', 'divider
            'Other': ['charity', 'donation', 'gift']
        }
        for cat, keys in defaults.items():
            c.execute("INSERT INTO Categories (category_name, keywords) VALUES (?, ?)", (cat, ",".join(keys)))

    conn.commit()
    conn.close()
```



```python
# NLTK Setup
try:
    nltk.data.find('corpora/stopwords')
except LookupError:
    nltk.download('stopwords', quiet=True)
try:
    nltk.data.find('tokenizers/punkt')
except LookupError:
    nltk.download('punkt', quiet=True)

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stop_words = set(stopwords.words('english'))
```

MILESTONE 3:

- Implemented Meta’s Prophet model to forecast future spending trends up to 365 days in advance.
- Designed advanced Visualizations to display forecast confidence intervals, seasonal trends, and weekly spending patterns.

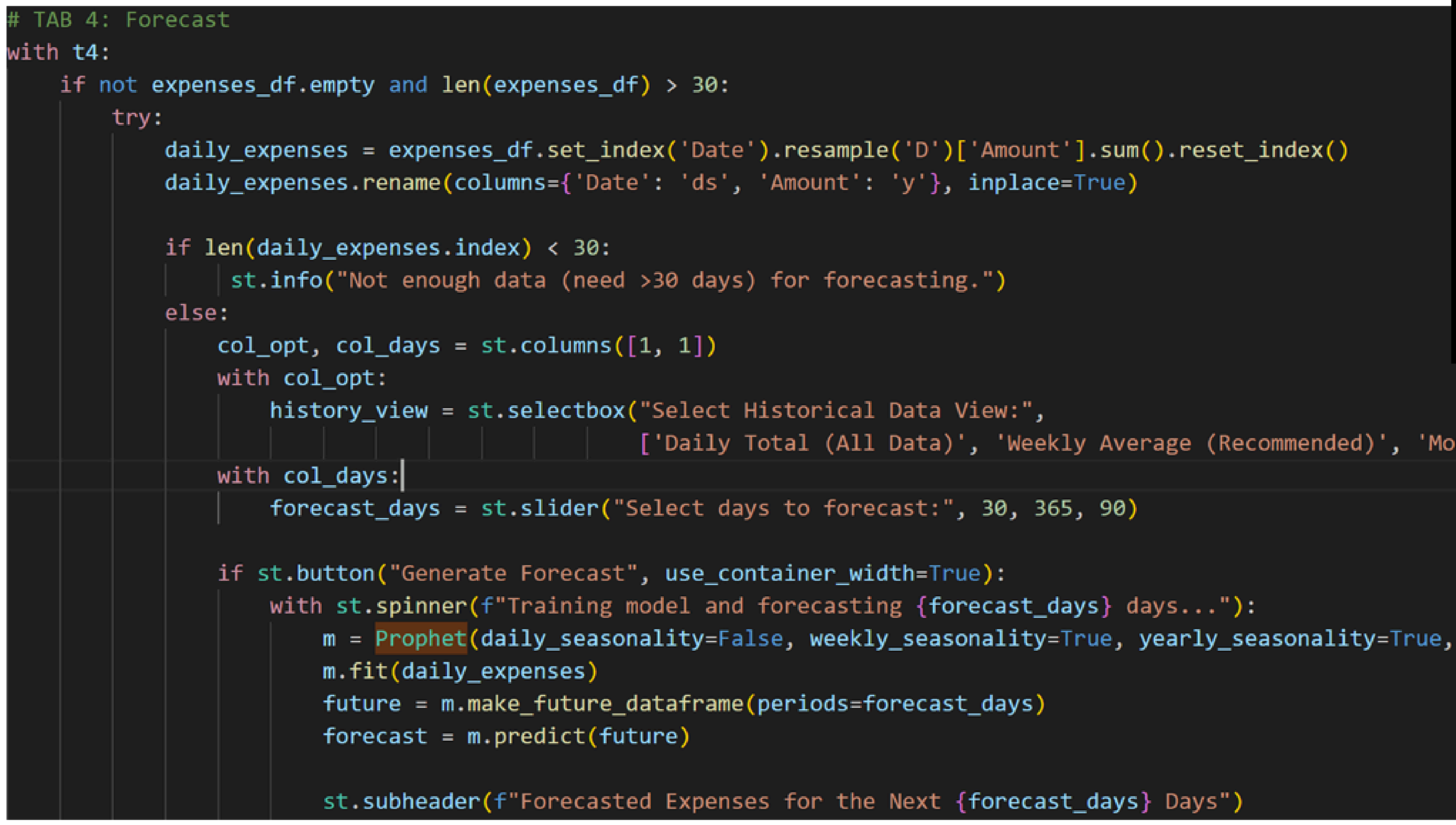


Select Historical Data View: Weekly Average (Recommended)

Select days to forecast: 90

Forecasted Expenses for the Next 90 Days

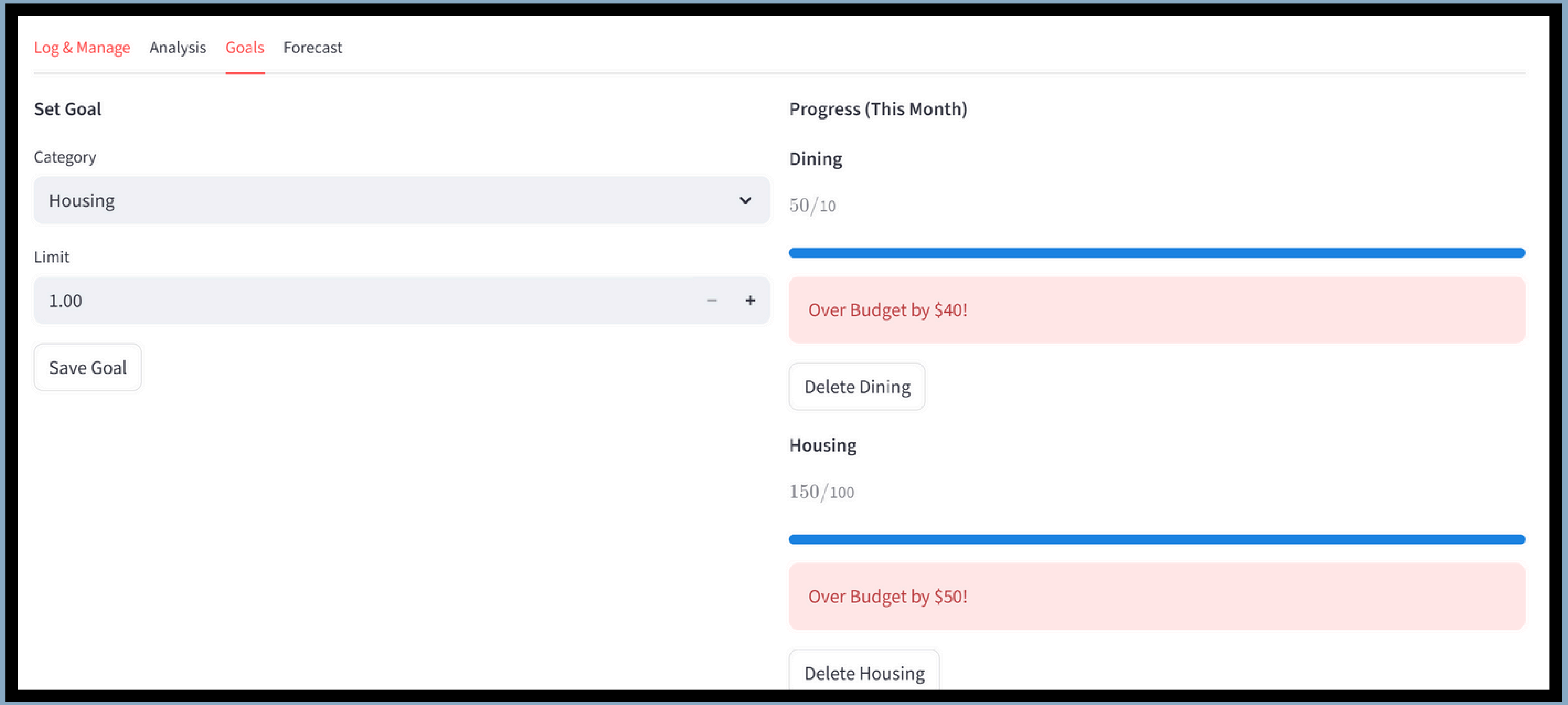
Legend: Black=Actual Weekly, Blue=Prediction, Area=Confidence



```
# TAB 4: Forecast
with t4:
    if not expenses_df.empty and len(expenses_df) > 30:
        try:
            daily_expenses = expenses_df.set_index('Date').resample('D')['Amount'].sum().reset_index()
            daily_expenses.rename(columns={'Date': 'ds', 'Amount': 'y'}, inplace=True)

            if len(daily_expenses.index) < 30:
                st.info("Not enough data (need >30 days) for forecasting.")
            else:
                col_opt, col_days = st.columns([1, 1])
                with col_opt:
                    history_view = st.selectbox("Select Historical Data View:",
                                                ['Daily Total (All Data)', 'Weekly Average (Recommended)', 'Monthly Average', 'Yearly
                with col_days:
                    forecast_days = st.slider("Select days to forecast:", 30, 365, 90)

                if st.button("Generate Forecast", use_container_width=True):
                    with st.spinner(f"Training model and forecasting {forecast_days} days..."):
                        m = Prophet(daily_seasonality=False, weekly_seasonality=True, yearly_seasonality=True, changepoint_prior_scal
                        m.fit(daily_expenses)
                        future = m.make_future_dataframe(periods=forecast_days)
                        forecast = m.predict(future)

                        st.subheader(f"Forecasted Expenses for the Next {forecast_days} Days")
```

MILESTONE 4:

- Built a Goal Setting Module allowing users to set monthly category limits with real-time "Over Budget" alerts and progress tracking.
- Deployed a dedicated Admin Dashboard for system monitoring and dynamic category management, allowing instant updates to the AI logic.



Set Goal

Category: Housing

Limit: 1.00

Progress (This Month)

Dining

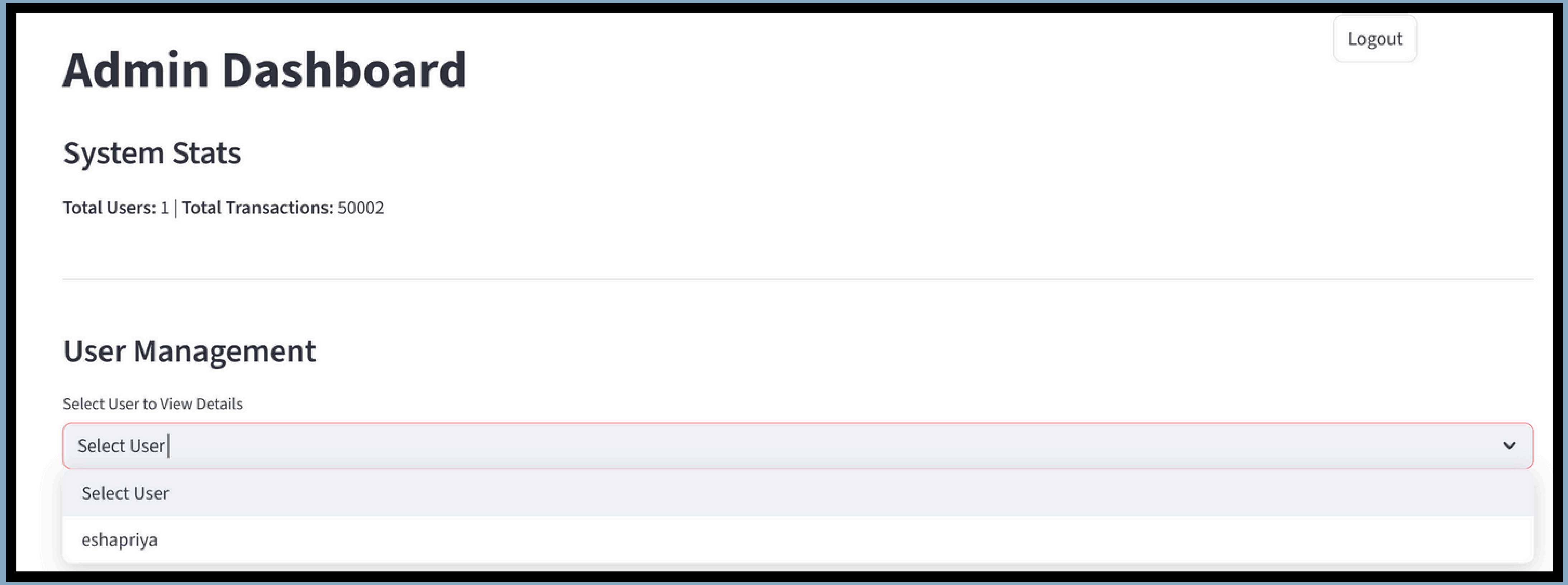
50/10

Over Budget by $40!

Housing

150/100

Over Budget by $50!



Admin Dashboard

System Stats

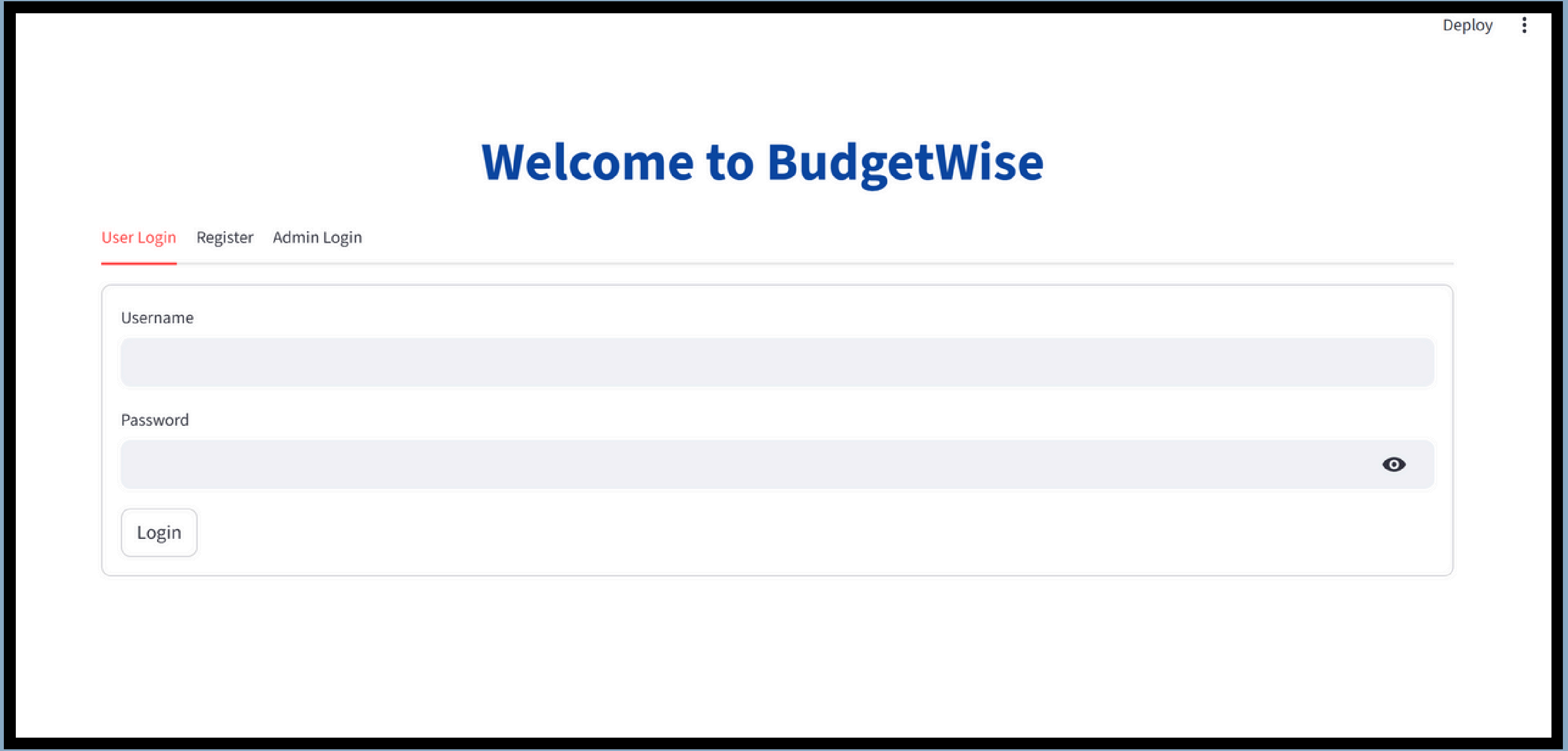
Total Users: 1 | **Total Transactions:** 50002

User Management

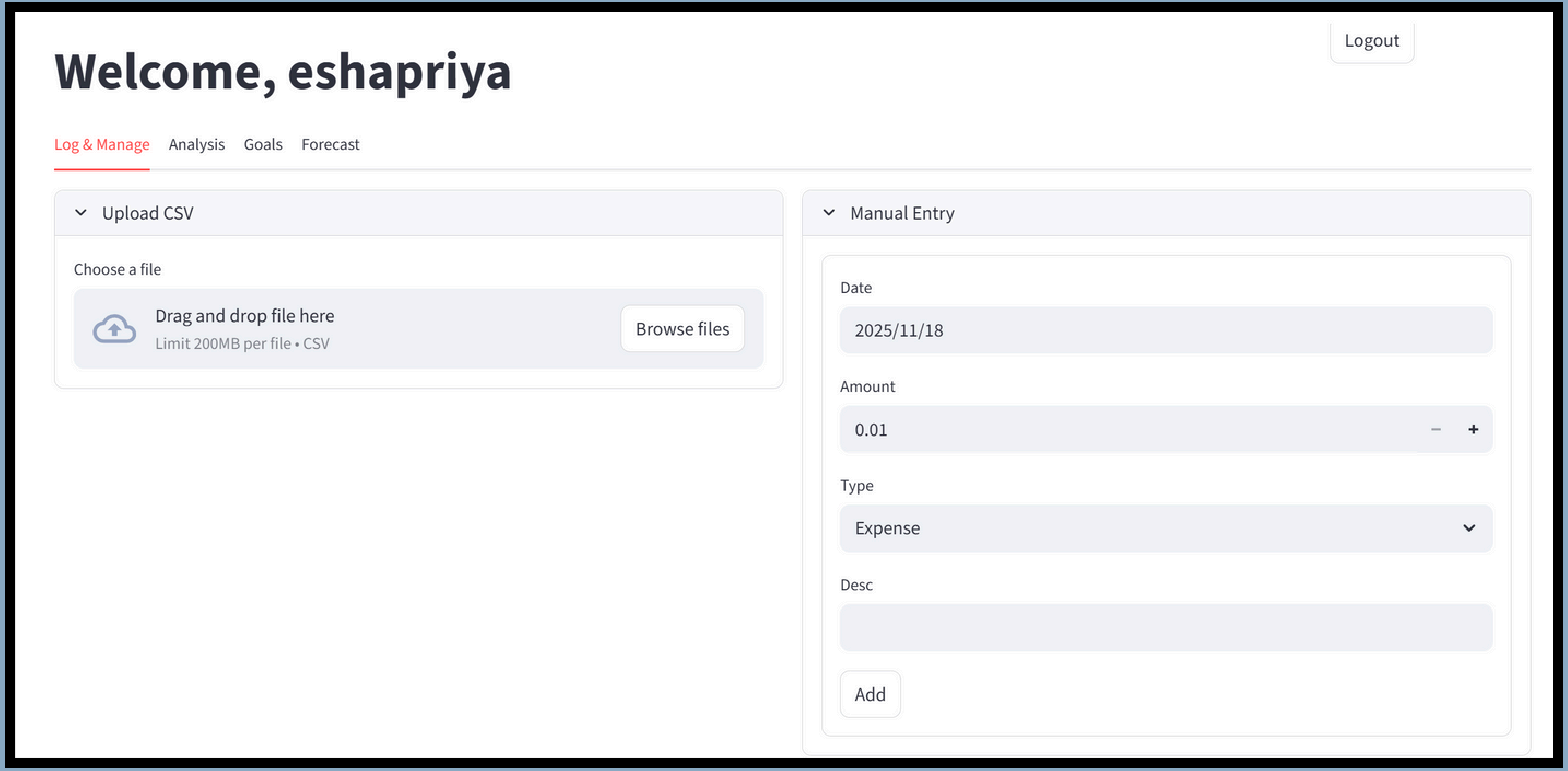
Select User to View Details: Select User

SNAPSHOTS OF THE PRODUCT

BUDGETWISE



Welcome to BudgetWise



Welcome, eshapriya

Date: 2025/11/18

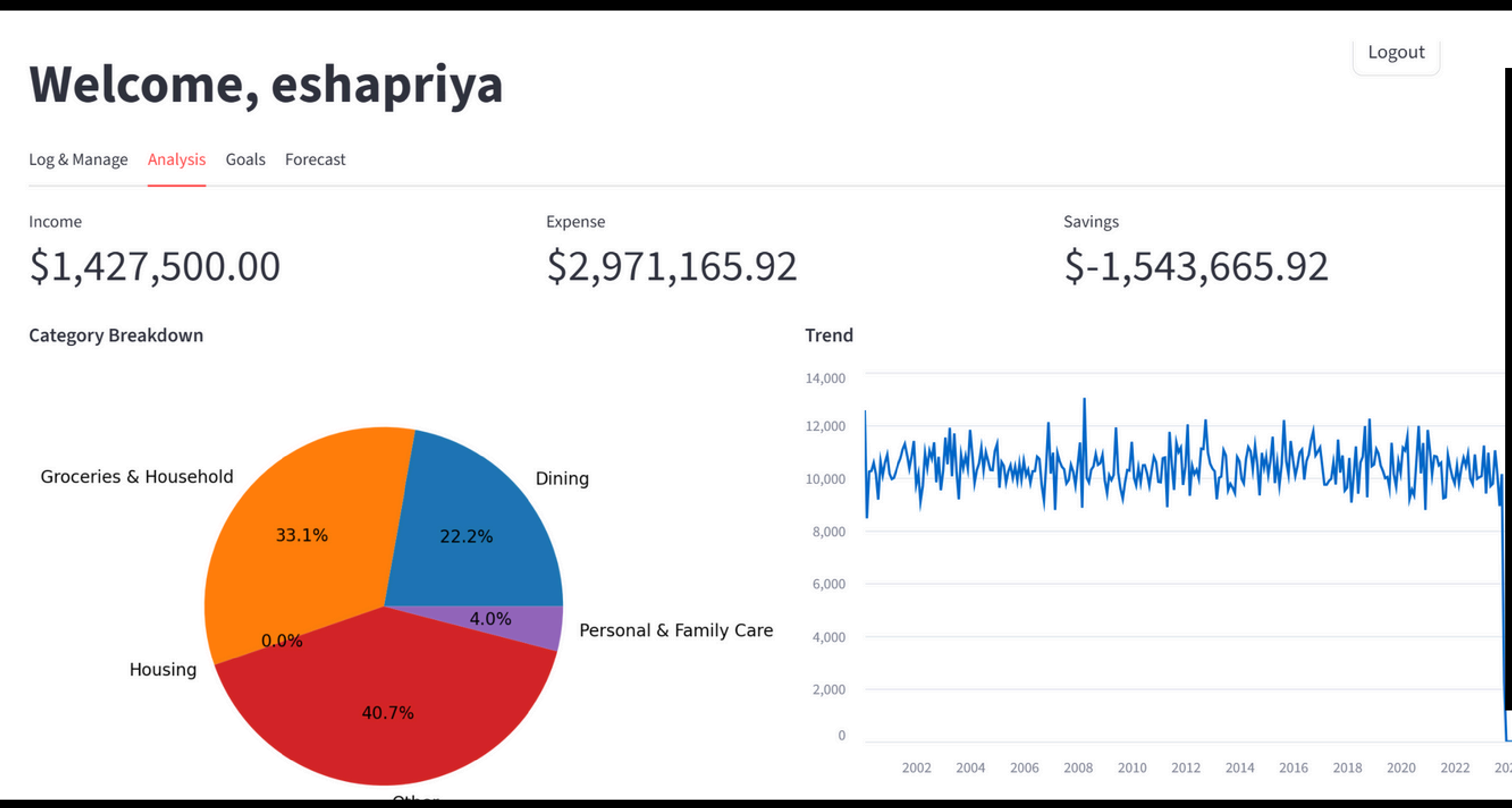
Amount: 0.01

Type: Expense

Desc:

User Login

BUDGETWISE



Logout

Welcome, eshapriya

Log & Manage | Analysis | Goals | Forecast

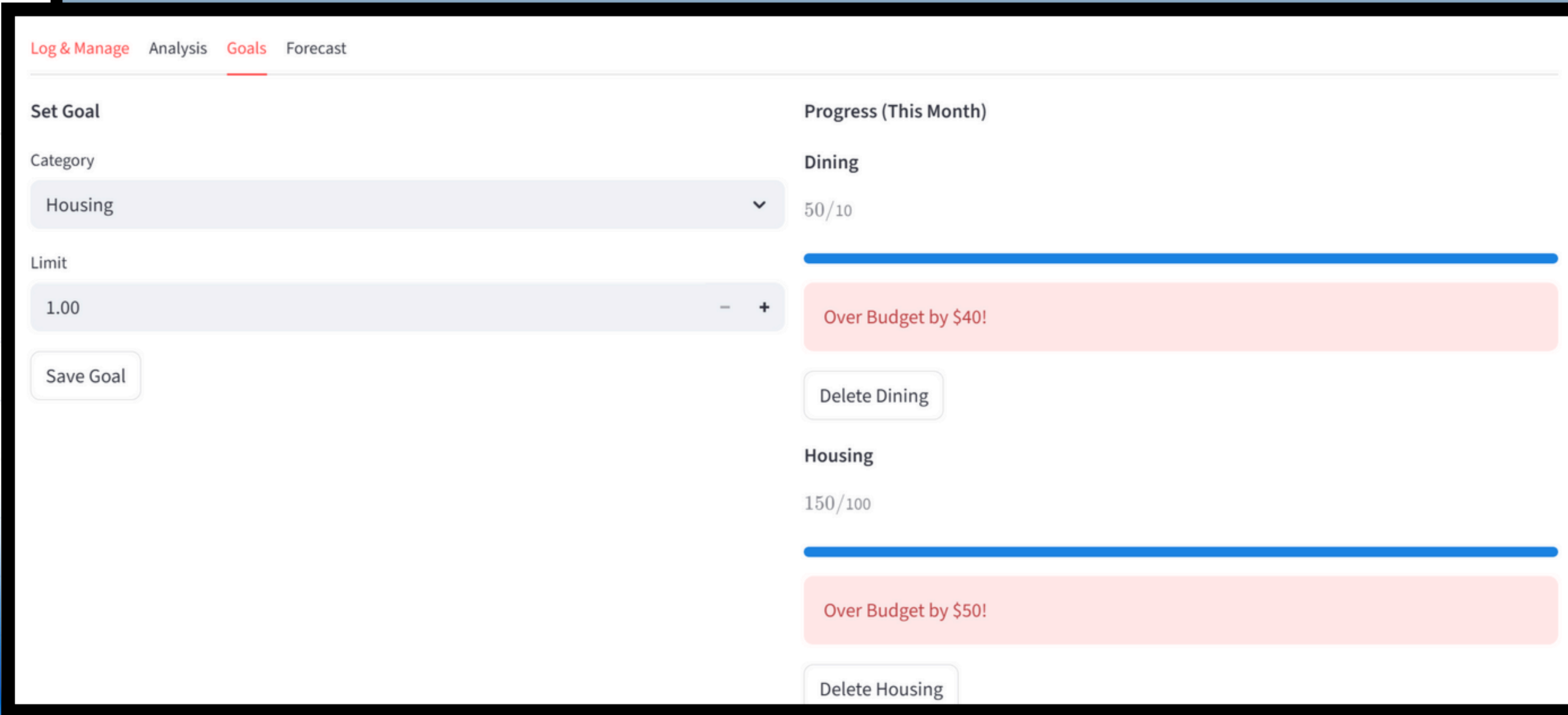
Income
$1,427,500.00

Expense
$2,971,165.92

Savings
$-1,543,665.92

Category Breakdown

Trend



Log & Manage | Analysis | Goals | Forecast

Set Goal

Category: Housing

Limit: 1.00

Save Goal

Progress (This Month)

Dining

50/10

Over Budget by $40!

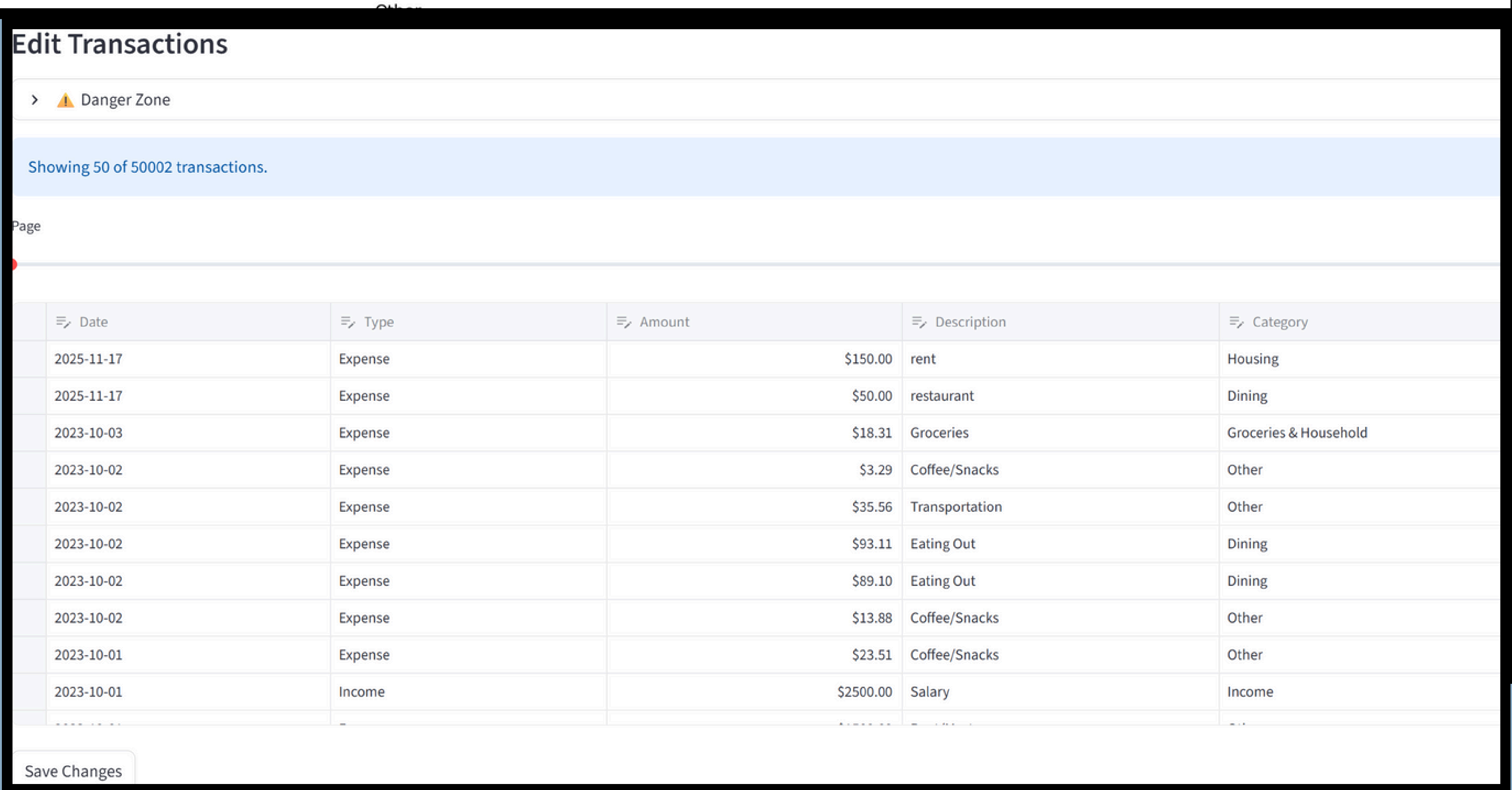
Delete Dining

Housing

150/100

Over Budget by $50!

Delete Housing



Edit Transactions

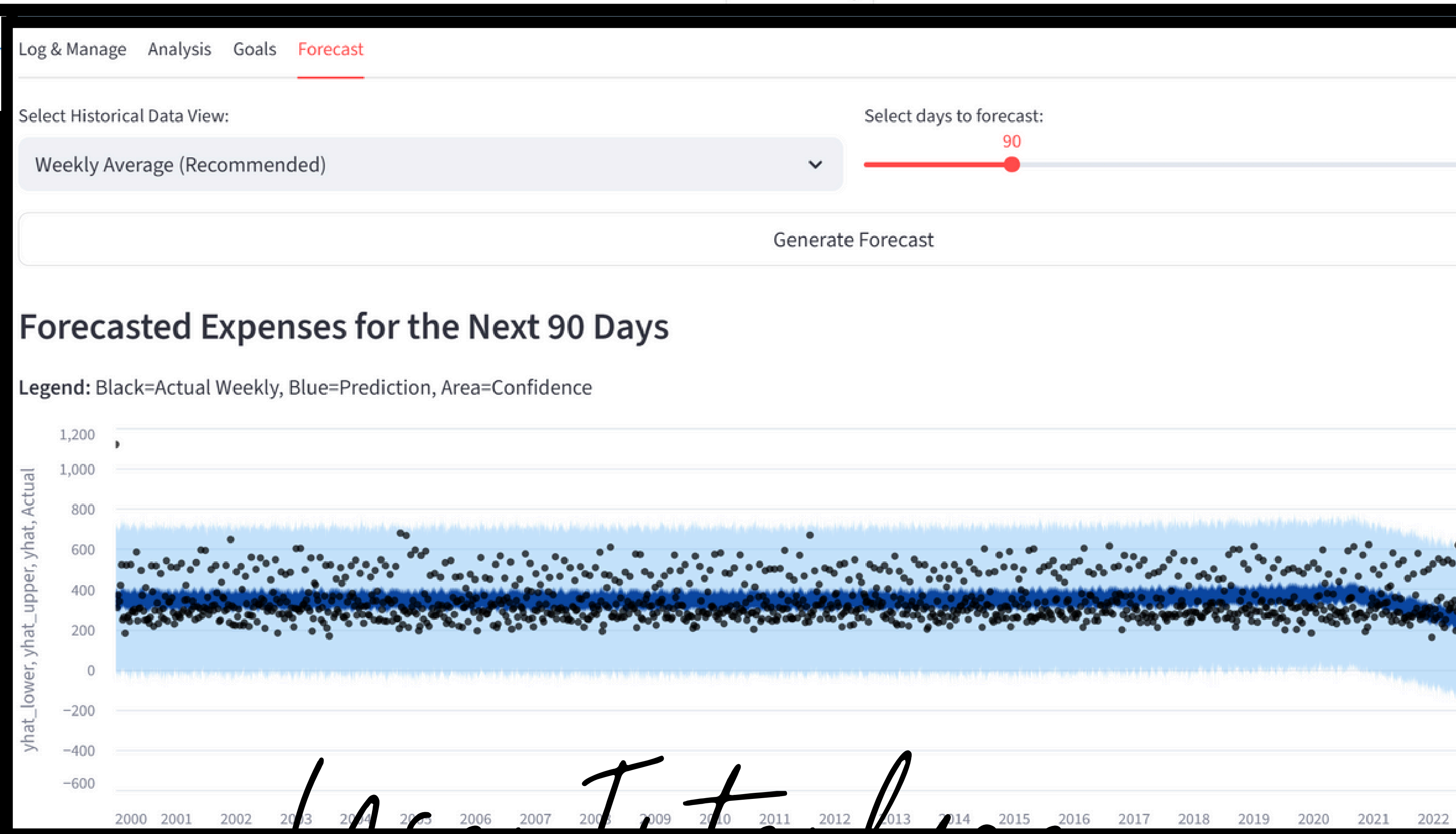
› ⚠ Danger Zone

Showing 50 of 50002 transactions.

Page

Date	Type	Amount	Description	Category
2025-11-17	Expense	$150.00	rent	Housing
2025-11-17	Expense	$50.00	restaurant	Dining
2023-10-03	Expense	$18.31	Groceries	Groceries & Household
2023-10-02	Expense	$3.29	Coffee/Snacks	Other
2023-10-02	Expense	$35.56	Transportation	Other
2023-10-02	Expense	$93.11	Eating Out	Dining
2023-10-02	Expense	$89.10	Eating Out	Dining
2023-10-02	Expense	$13.88	Coffee/Snacks	Other
2023-10-01	Expense	$23.51	Coffee/Snacks	Other
2023-10-01	Income	$2500.00	Salary	Income

Save Changes



Log & Manage | Analysis | Goals | Forecast

Select Historical Data View: Weekly Average (Recommended)

Select days to forecast: 90

Generate Forecast

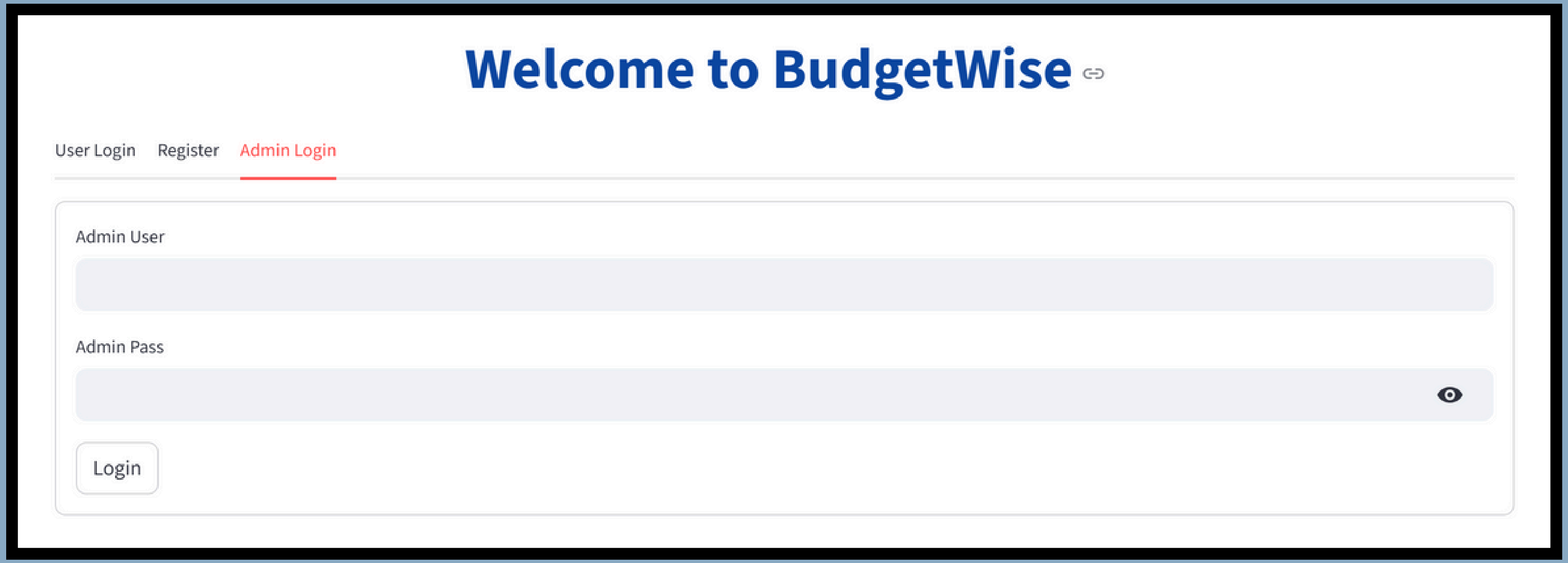
Forecasted Expenses for the Next 90 Days

Legend: Black=Actual Weekly, Blue=Prediction, Area=Confidence

User Interface...

BUDGETWISE

SNAPSHOTS OF THE PRODUCT



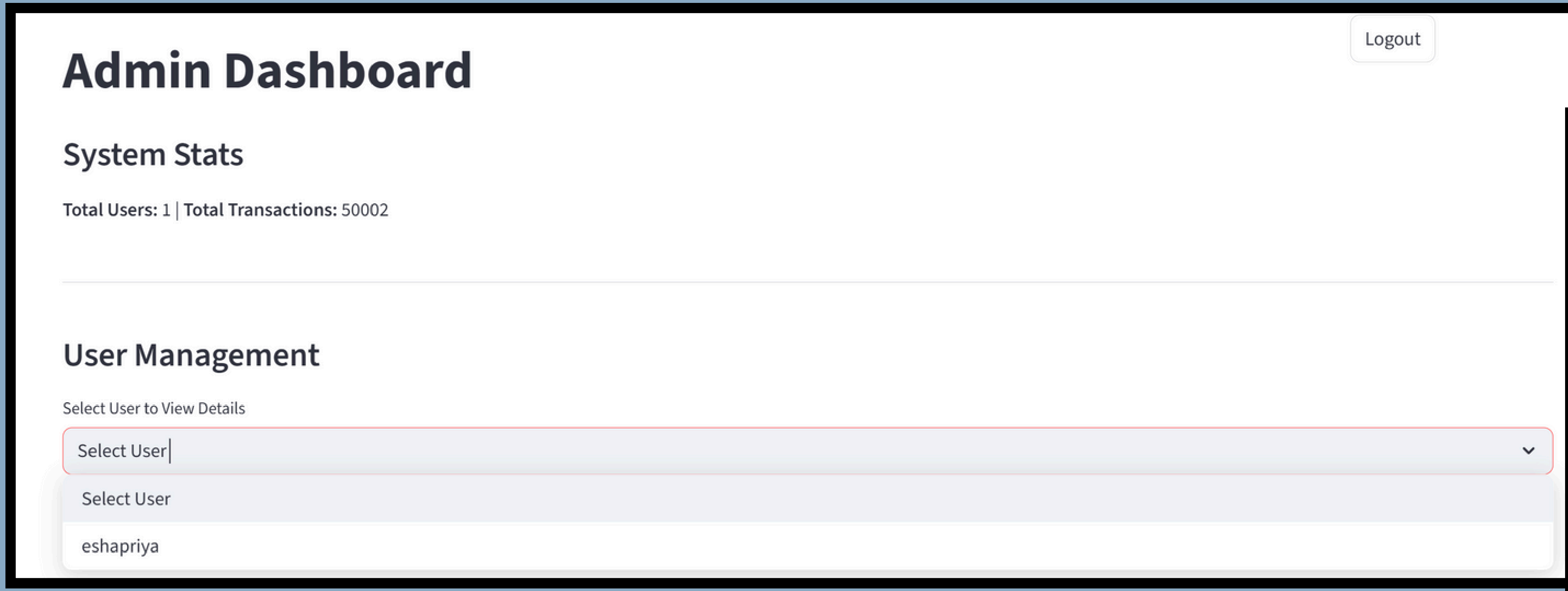
Welcome to BudgetWise

Admin User

Admin Pass

Admin Login

Admin Interface



Admin Dashboard

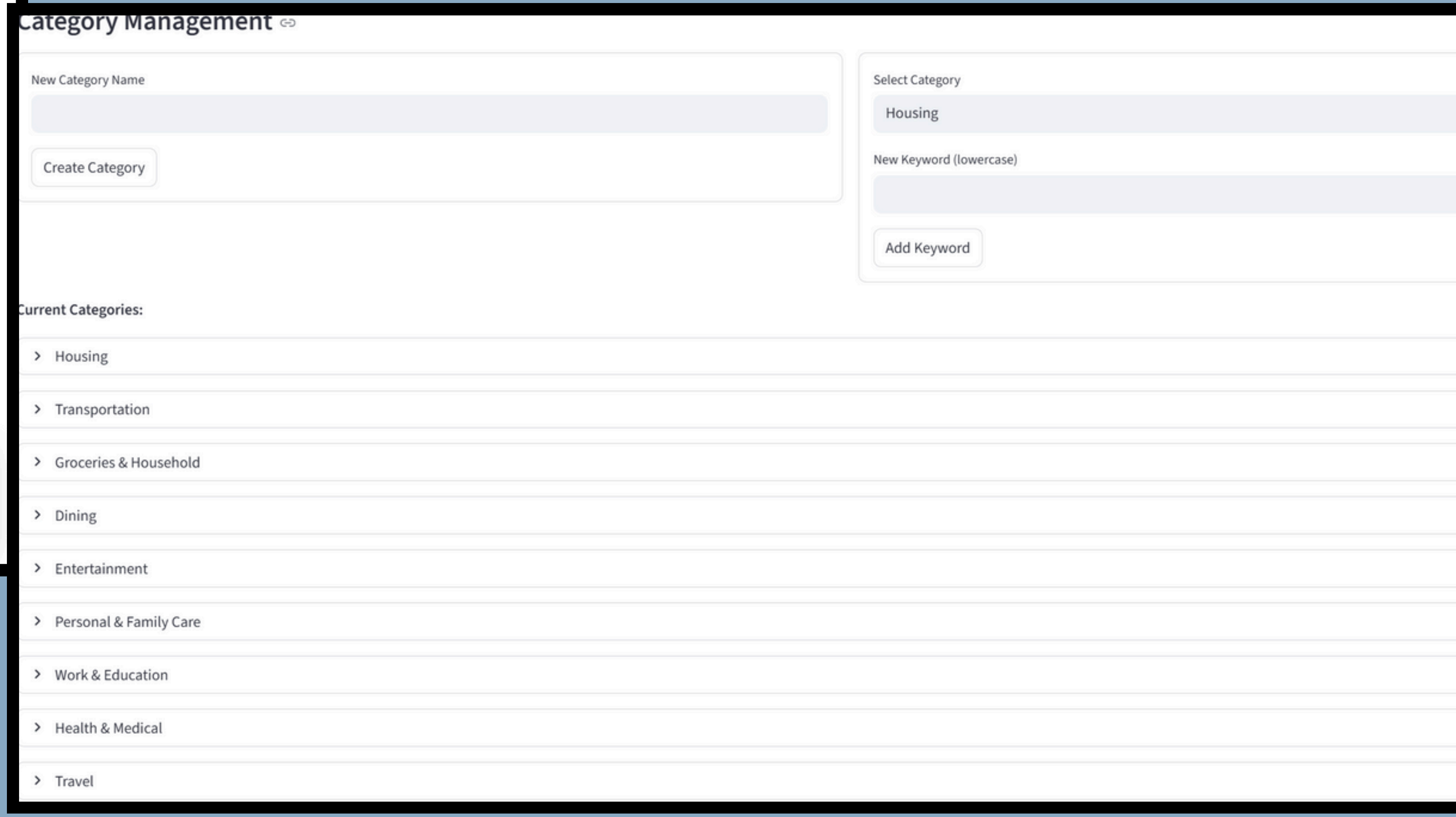
System Stats

Total Users: 1 | **Total Transactions:** 50002

User Management

Select User to View Details

Select User



Category Management

New Category Name

Select Category: Housing

New Keyword (lowercase)

Current Categories:

- Housing
- Transportation
- Groceries & Household
- Dining
- Entertainment
- Personal & Family Care
- Work & Education
- Health & Medical
- Travel

LINKS TO YOUR:

1. GitHub Public Repository : **BudgetWise Repository**

Thank you