WealthWave: Enhancing Personal Finance Management through Cloud Computing and AI

Tanmay Anil Rathi, Sai Aravind Yanamadala, Siri Chandana Errabelli, and Vamshi Naik Vislavath
New York University Tandon, tr2452@nyu.edu, sy3902@nyu.edu, se2596@nyu.edu, vv2289@nyu.edu

***Abstract* - WealthWave leverages cloud computing and advanced natural language processing to revolutionize personal finance management. Integrating AWS services with GPT's AI capabilities, it offers a cutting-edge approach to automate bill digitization and enhance financial analysis. This project addresses challenges like manual bill entry and inadequate expense insights, providing a seamless, intelligent solution. By combining cloud technology and large language models, WealthWave sets a new standard for intuitive, data-driven financial tools, demonstrating the extensive potential of these technologies in practical, impactful applications.**

Index Terms - AWS Services, Financial Data Analysis, Natural Language Processing, Personal Finance Management.

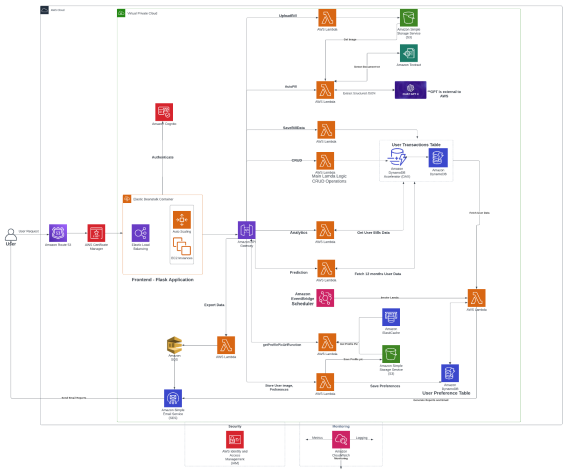
Problem Statement

A key challenge in personal finance management is not just tracking expenses but also effectively categorizing them to gain insights into spending patterns. Traditional financial tools often lack the sophistication needed for accurate expense categorization, leaving users with minimal understanding of their spending habits. This gap in functionality makes it difficult for individuals to analyze their expenses in meaningful categories, thereby hindering effective financial planning and budgeting. WealthWave addresses this critical need by offering an advanced solution for automated expense tracking and intelligent category inference, providing users with valuable insights into their financial behavior.

Motivation

WealthWave is motivated by the need to enhance personal finance management through technological innovation. As financial transactions become increasingly digital and diverse, traditional tools fall short in providing comprehensive and insightful analysis. WealthWave leverages cloud computing and AI, including GPT's language processing capabilities, to bridge this gap. It offers an intelligent, automated system for digitizing bills, categorizing expenses, and delivering actionable financial insights. This project aims to transform the tedious task of manual bill management into an efficient, insightful experience, empowering users to make informed financial decisions and fostering better financial awareness and stability.

Design and Architecture



WealthWave's architecture showcases a comprehensive use of AWS services, aligning with modern cloud computing practices for robust and scalable personal finance management. Key components and their roles are:

- **AWS Lambda**
  Use Case: Processes data and interfaces with DynamoDB. Chosen for its scalability and ability to manage high volumes of data processing without the need for server management.

- **Amazon S3**
  Use Case: Stores bill images and user profile pictures. Chosen for its high durability and ease of integration with other AWS services, ensuring secure and accessible data storage.

- **Amazon Textract**
  Use Case: Extracts text from bill images. Opted over GPT-Vision for cost efficiency and seamless integration within the AWS ecosystem. The extracted text is then processed by GPT language models for detailed insights and structured data.

- **Amazon API Gateway**
  Use Case: Manages API calls and throttles requests to control usage. This feature is crucial for managing costs associated with LLM API requests, limiting users to a specific number of bill data auto-fills per day.

- **Amazon DynamoDB**
  Use Case: Stores user data, user preferences and transaction records. DynamoDB's fast, efficient NoSQL capabilities make it ideal for handling the structured data generated from bill processing. Integrated with Amazon DynamoDB Accelerator (DAX) for faster data retrieval, enhancing the application's responsiveness.

- **Elastic Load Balancing and Auto Scaling**
  Use Case: Manages incoming user traffic and scales resources as needed. This choice ensures that WealthWave remains responsive and efficient under varying loads.

- **Flask Web Framework**
  Use Case: Serves as the backbone for the WealthWave web application. Flask was selected for its flexibility and ease of use, allowing for rapid development and deployment of the web service.

- **GPT Integration**
  Use Case: Provides AI-driven insights and categorization of financial data. The use of GPT's language processing capabilities enhances the application's ability to offer intelligent, context-aware financial analyses.

- **Amazon ElastiCache (Redis):** Enhances the speed and efficiency of retrieving frequently accessed data, such as user profile pictures in WealthWave. Selected for its high-performance, in-memory caching capabilities, ElastiCache significantly reduces the latency of data retrieval by storing key data in a readily accessible format. This is particularly beneficial for improving the load times of dynamic content on the application, enhancing the overall user experience. The integration with other AWS services, like Amazon S3 for storing profile pictures, ensures a streamlined, efficient architecture within the AWS ecosystem.

- **Amazon Cognito**
  Use Case: Manages user authentication and authorization. Cognito was chosen for its robust security features and ease of integration with other AWS services.

- **Amazon Route 53 and AWS Certificate Manager**
  Use Case: Handles domain management and SSL/TLS certification. The choice was influenced by the need to provide a secure HTTPS endpoint, a requirement for Amazon Cognito integration, ensuring user authentication is handled securely.

- **Amazon EventBridge Scheduler**
  Use Case: Automates the sending of email reports based on user preferences. In WealthWave, EventBridge triggers emails at frequencies set by users in the User Preference Table, ensuring timely financial report delivery.

- **Amazon SQS**
  Use Case: Handles data export requests from users. When a user requests data export, SQS queues the message, which is then processed to send the requested transaction data via email, using Amazon SES for reliable email delivery.

- **Amazon Simple Email Service (SES)**
  Use Case: SES is the primary service for sending emails within WealthWave. It's utilized for delivering exported data and scheduled financial reports to users. This service was chosen for its reliability and capability to handle high-volume email sending, ensuring users receive their requested data and reports in a timely and secure manner.

- **AWS Secrets Manager**
  Use Case: Incorporated AWS Secrets Manager to securely store and manage sensitive information, such as the OpenAI API key. This approach ensures that the API key is not hard-coded into the application, enhancing security. Secrets Manager provides a secure method to retrieve the key as needed without exposing it in the application's code or configuration files.

This architecture not only supports WealthWave's functionalities like bill scanning and expense analysis but also demonstrates efficient use of cloud resources and microservices architecture, considering the project's specific needs for security, scalability, and dynamic content delivery.

Implementation

The development of WealthWave involved setting up and configuring a suite of AWS services. This phase

included the establishment of a secure and scalable cloud environment, integrating services like Lambda, DynamoDB, S3, and Textract. Special attention was given to ensuring these services worked in harmony, providing a seamless backend infrastructure.

WealthWave's web application, developed using the Flask framework, constitutes the user-facing component of the project. The development process encompassed both frontend and backend development, ensuring a responsive and intuitive user interface. The application was then deployed using Elastic Beanstalk, which was chosen for its ability to efficiently manage the deployment, scaling, and handling of the web application.

- The Flask application was developed with a dual focus on user interface (frontend) and server-side logic (backend).
- The frontend was designed for ease of use, enabling users to upload bills, view analytics, and set preferences.
- Developed a user-friendly interface for bill uploading, data visualization, and user settings management.
- Integrated AWS Cognito for secure user authentication and session management.
- **Deployment Considerations:** Deployment on AWS Elastic Beanstalk was chosen for its ability to manage and scale the Flask application automatically, handle traffic efficiently, and integrate seamlessly with other AWS services.

During the implementation, several challenges were encountered, such as ensuring efficient communication between different AWS services and optimizing the application's performance. Solutions included the use of Amazon SQS for reliable message queuing and EventBridge Scheduler for automated task management.

- **AWS Environment Setup:** The foundation of WealthWave's backend was established in AWS. This involved creating and configuring services like Lambda for serverless computing, DynamoDB for database needs, and S3 for storage. Special focus was on setting up Textract for its OCR capabilities.
- **Service Configuration:** Each service was meticulously configured to ensure seamless inter-service communication and data flow. Configurations were tailored to optimize performance and cost, with attention to security and scalability.
- **Integration Strategy:** The integration strategy involved ensuring that the AWS services communicated effectively with the Flask application. This included setting up API Gateway endpoints for various functionalities like data retrieval, bill processing, and user preference management.

We conducted extensive testing under various scenarios to ensure reliable performance, including different bill formats and light conditions. Iteratively refined the application based on test results, ensuring a seamless user experience.

WORKFLOWS

I. Landing Page Workflow

The Landing Page Workflow is a crucial part of the WealthWave application, serving as the gateway for user interaction. It's where first impressions are formed and initial user engagement occurs. This workflow is meticulously designed to be inviting and intuitive, guiding users smoothly from the point of entry to deeper engagement with the application.

Now, let's delve into the specifics of this workflow:

- **Initial Presentation:**
  The landing page of WealthWave, hosted at https://wealthwavenyu.net/, is designed to be user-friendly and informative, providing a clear understanding of what WealthWave offers.
- **Domain Usage:**
  The use of a custom domain, wealthwavenyu.net, adds professionalism and brand identity, assuring users they are on an official and secure platform.
- **Sign-In/Up Integration:**
  Featuring a 'Sign In / Sign Up' button, the page directs users seamlessly to the AWS Cognito sign-in page for secure authentication.
- **User Experience:**
  The page layout is optimized for a positive user experience, with straightforward navigation and a design that highlights WealthWave's key features and benefits.

This workflow is designed to efficiently transition users from initial interest to active engagement with the WealthWave platform.

II. Sign In and Sign Up Workflow Using AWS Cognito

The "Sign In and Sign Up" workflow using AWS Cognito is a critical component of WealthWave, ensuring secure and

streamlined user access. This process involves the integration of AWS Cognito, a robust authentication service, which manages user identities and facilitates secure sign-in and registration. The workflow is designed to be user-friendly, ensuring that both new and returning users can easily navigate the authentication process.

Now, let's delve into the specific steps of this workflow.

- **User Action:** The user clicks the 'Sign In / Sign Up' button on WealthWave's landing page.
- **Cognito Redirection:** They are redirected to AWS Cognito's authentication UI.
- **Registration/Log In:** New users register, while returning users log in with their credentials.
- **Authentication Confirmation:** Cognito verifies user credentials and manages the authentication process.
- **Successful Sign-In:** Once authenticated, users are redirected back to WealthWave, typically to their dashboard.
- **Session Initiation:** WealthWave initiates a user session, providing a personalized experience based on the authenticated user profile.

This workflow is integral to ensuring a secure user experience, leveraging AWS Cognito's robust authentication mechanisms.

III. Upload Bill Workflow in WealthWave

The "Upload Bill" workflow in WealthWave is a pivotal feature, enabling users to upload their financial documents for analysis. This process is designed to be user-friendly, allowing users to easily upload bill images and receive immediate financial insights.

- **User Uploads Bill:** Users select and upload a bill image through the application's interface.
- **Image Processing:** The image is processed using Amazon Textract, which extracts text from the bill image.
- **OpenAI LLMs:** The text extracted from textract is given to ChatGPT using API as a user message. The custom prompt is specified as a system message that returns financial data like date, total amount and category (Provided in prompt) as the response in JSON format.
- **Data Review:** Extracted data from the JSON response is displayed to the user for verification and any necessary corrections.
- **Confirmation and Storage:** Users confirm the accuracy of the extracted data, which is then saved to the system for future reference and analysis.

This workflow highlights WealthWave's commitment to simplifying personal finance management, leveraging advanced cloud computing technologies for efficient data processing.

IV. Analytics Workflow in WealthWave

The "Analytics" workflow in WealthWave plays a crucial role in providing users with insightful data analysis of their financial habits. This process helps users understand their spending patterns and make informed financial decisions.

- **Data Aggregation:** Upon accessing the analytics section, the application aggregates user's financial data.
- **Analysis and Categorization:** Using the integrated AWS services, the application analyzes the transactions, categorizing them for better understanding.
- **Visualization:** The analyzed data is then presented in an easy-to-understand format, often through charts or graphs, allowing users to visualize their spending patterns.
- **User Interaction:** Users can interact with the data, filtering or sorting it to gain different perspectives on their financial habits.

This workflow underscores WealthWave's capability to transform raw financial data into meaningful insights, aiding users in better financial management.

V. Predictions Workflow in WealthWave

The "Predictions" workflow in WealthWave is designed to provide users with forecasts of their future spending based on historical data. This predictive analysis is crucial for effective financial planning and management.

- **Historical Data Analysis:** The application analyzes the user’s past year’s financial data to identify spending trends.
- **Forecast Generation:** Utilizing linear regression model, the application predicts

future spending in various categories for the next month.

- **Presentation to User:** These predictions are then displayed to the user, often in a graphical format, enabling easy comprehension and planning.

- **User Interaction and Customization:** Users can interact with this data, customizing the view or adjusting parameters for different forecast scenarios.

This workflow demonstrates WealthWave's advanced use of predictive analytics, offering users valuable insights into their future financial trends.

VI. Export Data Workflow in WealthWave

The "Export Data" workflow in WealthWave is crucial for users who wish to download and keep a record of their financial transactions. This functionality provides a tangible way to manage and review financial data outside the app.

- **Selection of Date Range:** Users choose a specific date range for the data they wish to export.

- **Data Compilation:** The application compiles the requested transaction data for the specified period.

- **Email Delivery:** The compiled data, typically in a CSV format, is sent to the user's email address, likely using Amazon Simple Email Service (SES) for secure and efficient delivery.

- **User Notification:** Users are notified within the application that their data export request is being processed and will be delivered to their email.

This workflow provides a user-friendly method for data portability, enhancing the overall utility of WealthWave.

VII. Settings Workflow in WealthWave

The "Settings" workflow in WealthWave allows users to customize their profile and preferences, enhancing their personal experience with the application.

- **Accessing Settings:** Users navigate to the settings section of the app.

- **Profile Customization:** Users can update personal details like profile pictures, stored securely in Amazon S3.

- **Preference Adjustments:** Users set preferences, such as email frequency for reports, leveraging AWS services for backend processing.

- **Saving Changes:** Changes are saved and applied immediately, affecting how the user interacts with the application henceforth.

This workflow enables users to have a tailored experience, aligning WealthWave's functionalities with their individual needs.

VIII. Logout Workflow in WealthWave

The "Logout" workflow in WealthWave is designed to ensure users can securely exit their session, maintaining the integrity and security of their account.

- **Initiating Logout:** The user selects the logout option within the application.

- **Session Termination:** The application securely terminates the user's session, ensuring no residual data is accessible.

- **Redirect to Landing Page:** Post-logout, the user is redirected back to the landing page of WealthWave.

- **Security Assurance:** This process ensures that user data remains secure and that the session is completely closed.

This workflow is a crucial aspect of WealthWave's security measures, safeguarding user information by securely managing session termination.

Challenges Encountered and Resolutions

Throughout the development of WealthWave, the team faced and overcame several significant challenges, ensuring the project's success.

We encountered complexities in structuring bill data, resolved by opting for Textract and appropriate LLMs for reliable data extraction. Faced initial hurdles in integrating AWS Cognito and Elastic Beanstalk, resolved by purchasing a domain and configuring AWS Certificate Manager for HTTPS requirements.

- **Data Structuring:** Initially faced challenges with structuring bill data. This was overcome by using Amazon Textract for high-quality text extraction and integrating it with less complex LLMs for data analysis.

- **Authentication Integration:** Integrating AWS Cognito for secure user authentication presented challenges, especially in hosting the application in a scalable environment. The solution involved purchasing a domain and configuring AWS Certificate Manager to meet HTTPS requirements, a necessary step for Cognito's secure operation.

- **DNS and Hosting Adjustments:** DNS changes for the new domain took time to propagate, causing delays. Patiently managed DNS settings and awaited propagation to ensure that the domain functioned correctly, particularly for Cognito's HTTPS callbacks.

- **Bill Image Processing Under Varying Conditions:** Processing bill images captured in low light or at odd angles was a challenge. Leveraged Amazon Textract's robust OCR capabilities, which proved effective in handling images under varied conditions, ensuring reliable data extraction.

- **Cost Management with LLM APIs:** We had to manage the cost associated with using Large Language Model (LLM) APIs for data processing. We had paid 5$ for the OpenAI API, which can be a limiting factor even for a large-scale app, as the amount should be kept under a limit by restricting the user requests after a threshold. We implemented throttling and rate limits on the API Gateway to control usage, ensuring efficient management of expenses associated with LLM APIs.

- **Integration of Amazon QuickSight:** We wanted to make use of the AWS ecosystem of tools including Quicksight for analytics page but we encountered difficulties embedding Amazon QuickSight dashboards in the application. The primary issue was the limited lifespan of embedded URLs, which expired after a certain period. This limitation, coupled with limited reference materials and tutorials, made it challenging to integrate QuickSight seamlessly into WealthWave for real-time analytics visualization. We used python libraries as a workaround.

- **Integrating Predictive Analytics:** Incorporating predictive analytics for future expense forecasting was initially planned using some AWS Services like forecast but that turned out to be tricky. We therefore shifted our approach by utilized machine learning techniques and integrated them with the application's backend, possibly using AWS Lambda for computation.

Future Enhancements

As WealthWave continues to evolve, several enhancements are envisioned to further improve its capabilities and user experience:

- **Advanced AI Integration:** Incorporating more sophisticated AI algorithms for deeper financial analysis and personalized recommendations.

- **User Interface Improvements:** Continuously refining the user interface for greater intuitiveness and accessibility, including mobile optimization.

- **Expanded Financial Tools:** Introducing a broader range of financial management tools, such as budget planning and goal setting features.

- **Real-Time Data Processing:** Enhancing the application's ability to process and display data in real-time for immediate insights.

- **Community Features:** Adding community-driven functionalities, like forums or advice sections, for users to share financial tips and experiences.

- **Enhanced Security Measures:** Implementing stronger security protocols and privacy safeguards as the application scales and handles more sensitive data.

These potential enhancements aim to make WealthWave not only a tool for financial management but a comprehensive platform for financial wellness and community engagement.

References

[1] Amazon Web Services, Inc., "AWS Documentation," AWS Documentation, [Online]. Available: https://docs.aws.amazon.com/. [Accessed: Jan 28, 2023].

[2] Amazon Web Services, Inc., "Amazon Cognito," Amazon Cognito, [Online]. Available: https://aws.amazon.com/cognito/. [Accessed: Jan 28, 2023].

[3] Amazon Web Services, Inc., "Amazon Elastic Beanstalk," Amazon Elastic Beanstalk, [Online]. Available: https://aws.amazon.com/elasticbeanstalk/. [Accessed: Jan 28, 2023].

[4] Amazon Web Services, Inc., "AWS Certificate Manager," AWS Certificate Manager, [Online]. Available: https://aws.amazon.com/certificate-manager/. [Accessed: Jan 28, 2023].

[5] Amazon Web Services, Inc., "Amazon ElastiCache," Amazon ElastiCache, [Online]. Available: https://aws.amazon.com/elasticache/. [Accessed: Jan 28, 2023].