Icon

Description automatically generated with low confidence

|  |
| --- |
| PERSONAL EXPENSE TRACKER  2021 |
|  |
| July 13  Geethanjali College of Engineering & Technology  Authored by: Rohan Kausik Nandula |

# 1 Introduction

|  |
| --- |
| 1.1 **Overview** As part of GCET Internship program management for students , I got an opportunity to work on “**Personal Expense Tracker**” , a miniscule project , which provides an ability for end users to manage life cycle of various expenses incurred by them on a monthly basis. 1.2 **Purpose** Smart Internz group collaborated with GCET to envision an approach and provide students an opportunity to understand latest technologies and trends. Developing project would help students to understand current market requirements and enhance their abilities to design and deliver such solutions in future. |
| *“AWS Cloud based solution ‘APIfied’ along with Lambda functions and No SQL DB , UIX (front end using Python- Flask web framework) and docker for build /containerization/deployment was developed ”* |
| To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.  Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need. 2 Literature Survey2.1 Existing problem The problem with creating this application lies in the fact that user data representation and management lacked scalability. With the ever-growing user market, requirement management has become a crucial part of the running applications with ease. To overcome the restricted server space in managing the application, we turned to Amazon Web Services(AWS). 2.2 Proposed solution AWS allows application providers, ISVs, and vendors to quickly and securely host applications – whether an existing application or a new application. It enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon’s massive infrastructure, you have access to compute and storage resources when you need them. These features allowed me to be expressive and effective in the way I developed the application. 3 Theoretical Analysis3.1 Block Diagram: Diagrammatic overview of Use-Cases  3.2 Hardware / Software designing  Hardware and software requirements of the project:  Thanks to AWS services and Docker containerization, the hardware and software specifications required to develop and use the application are minimal. Here are the baseline hardware requirements:  Operating System: *Windows 8 onwards, MacOS High sierra onwards*  Processor Type: *Intel core i3 9th gen/ Intel core i5 7th gen/ Intel core i7 5th gen onwards*  Processor Speed: *2 GHZ*  Memory: *250 mb and above*  Here are the software requirements:  Technologies: AWS, Flask Python, Docker  Framework: Flask Framework V2.0.1  Backend end tools: PyCharm V2020.3, AWS Dynamo DB, AWS API, AWS Lambda, AWS SNS, AWS IAM  Front end tools: HTML,CSS, JavaScript  Support Tools: Notepad++, Sublime Text editor 4 Experimental Investigations *Analysis or the investigation made while working on the solution:*  "# Personal-expense-tracker" PROJECT CHECKPOINTS/ANALYSIS   * Application source file ---- main.py   Following functionalities have been developed and tested: a. Login b. User registration c. Simple dashboard with wallet balance d. Adding expenses e. Updating wallet balance f. Home page  APIs developed: (Used to access a Lambda function through a link )  a. Wallet balance retrieval API: <https://ss979hyehb.execute-api.ap-south-1.amazonaws.com/WalletBalance>  b. Wallet balance updation API: <https://l1gdzvb6k6.execute-api.ap-south-1.amazonaws.com/addwallamount>  c. API to set the Expenses: <https://ket0h58q15.execute-api.ap-south-1.amazonaws.com/ExpenseAPI>?  d. API to fetch the Expenses: <https://5996kr662d.execute-api.ap-south-1.amazonaws.com/ExpenseQA>?  e. API to fetch user details: <https://nirmgp3j2c.execute-api.ap-south-1.amazonaws.com/fetchuser?user=>  f. API to push user details: <https://nqwsosw3ag.execute-api.ap-south-1.amazonaws.com/QA>?  Associated Lambda functions:  a. putinfo: pushes user info into a database  b. fetchUser: pulls user details from the database  c. walletbalance : pulls wallet balance from the database  d. push\_expense\_data : pushes expense data into the database  e. fetchExpenses: fetches expense details of a user from the database  f. addwalletamount : pushes additional amount to wallet balance  g. limitsns(Created but not working): tried to send an SNS message upon going below a set wallet amount.  Associated AWS DynamoDBs:  a. users - user details  b. expense\_summary - expenses related to user c. wallet\_balance - contains user wallet balance d. monthly\_limits - contains user set monthly limits for expenses  5 FLOWCHART  Diagram showing the control flow of the solution:    Reports are shown in graph format.    6 RESULT  Final findings (Output) of the project along with screenshots:  Check out the application overview video here: <https://drive.google.com/file/d/1NDMh-e819fsGkIoorqc-Nq5cmYV71k2O/view?usp=drivesdk>  Screenshots of Backend services used:   1. Dynamo DB tables:      1. Users: 2. Expense\_Summary:      1. Wallet\_amount:      1. API’s used:      1. SNS Service:        1. EC2 Instance for Docker(runs the application publicly):     7 ADVANTAGES & DISADVANTAGES  List of advantages and disadvantages of the proposed solution:  Advantages:   1. Know where you're spending more than you actually think you are 2. Figure out ways to cut back on your spending 3. Know how much extra payments you can make towards your debt 4. Plan for future large purchases 5. Create a savings plan for putting money away every month 6. Help "pace" your monthly spending, stretching your dollar further   Disadvantages:   1. Lacks a monthly budget feature, which may prompt the user to spend aimlessly. 2. Your information is less secure, and probably being used and sold. 3. Automating everything to do with your finances can make you financially lazy. If your bills are paid automatically and your finances are track automatically, then what is there left for you to do? Not a lot, to be honest. So you might stop caring about what you’re spending and where your money is going. 4. Data maintenance may be difficult when it comes to handling user data. 5. Person who is handling system must have some computer knowledge. 6. Requires lot of effort to understand the project process flow.   8 APPLICATIONS  The areas where this solution can be applied:   1. This application is useful to people belonging to the middle and lower classes to maintain their wallets. 2. Small and upcoming businesses will find this app beneficial when it comes to maintaining a capital amount. 3. Event organizers with limited finance can use this app to keep track of their expenses and wallet balance. 4. Users belonging to many more professions can use this application for their benefit.   9 CONCLUSION  Finally, a summary of all my work(in a few points)   1. Keeps account holder information 2. Manages daily transactions of users 3. Provides notifications to user if the wallet balance goes below a preset limit 4. Very useful in visualizing your expenses 5. The EXPENSE GRAPH feature will include the date of transaction and also will include the category of expense. 6. Flexible and adaptive application which suitable to a normal person who is managing his/her daily expense and income. 7. It is also a powerful application for a business person who is handling many daily transactions.   10 FUTURE SCOPE  **Enhancements that can be made in the future:**   1. More functionalities like receipt capture, location settings, backup data are in our sights. 2. The project can be expanded as per the need of the users(through feedback). 3. UI developments to make this application more user friendly is something to look forward to. 4. Monthly budget feature to be included soon. 5. Planning to synchronize with bank account so upcoming bill alert should be mailed to the particular user.   **Prepared by:**   1. **Nandula Rohan Kausik – 19R11A0581 – CSE-B** 2. **Sirisanagandla Venu – 19R11A0591 – CSE-B** 3. **Bindesh Yadav – 19R11A0554 – CSE-B** 4. **G.Akhil – 19R11A0564 – CSE-B**   **As part of the Personal Expense Tracker App using fundamentals of Cloud Application development. Special thanks and gratitude to the following as part of completion of this externship program:**   * Mr.U.K.Susarla – Principal/Program Coordinator GCET * Mr. Sandeep Doodigani – SmartInternz instructor * Mr. N.Balasubramanyam – Chief Architect(TCS) – Overall Guidance |