Tribhuvan University
Institute of Science and Technology

Bhaktapur Multiple Campus

Doodhpati, Bhaktapur, Nepal

An Internship Report

On

“Software Engineer”

At

Khalti Pvt. Ltd.

Submitted by

Anukul Adhikari (T.U. Exam Roll No. 23226/076)

An Internship Report Submitted in partial fulfillment of the requirement of Bachelor of Science in Computer Science & Information Technology (BSc.CSIT) 8th Semester of Tribhuvan University, Nepal

July, 2024

MENTOR’S RECOMMENDATION

This is to recommend that **Anukul Adhikari** has submitted the internship report entitled **“An Internship report on Software Engineer at Khalti Pvt. Ltd.”** for the fulfillment of the requirement of the Bachelor’s degree of Computer Science and Information Technology (BSc. CSIT) is processed for the evaluation.

………………….

Er. Subit Raj Pokhrel

Mentor

Khalti Pvt. Ltd.

Bakhundole, Lalitpur

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this report has been prepared under my supervision by **Anukul Adhikari** entitled **An Internship Report on Software Engineer at Khalti Pvt. Ltd.** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology of Tribhuvan University is processed for the evaluation.

. . . .

Mr. Pratik Timilsina

Supervisor

Department of Computer Science and Information Technology

Bhaktapur Multiple Campus

Doodhpati, Bhaktapur

LETTER OF APPROVAL

This is to certify that this internship report prepared by **Anukul Adhikari** entitled **"An Internship report on Software Engineer at Khalti Pvt. Ltd"** has been submitted to the Department of Computer Science for acceptance in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology. In our opinion, it is satisfactory in the scope and quality as a project for the required degree

Signature of Supervisor . . . . **Mr. Pratik Timilsina** Bhaktapur Multiple Campus	**Signature of Mentor** . . . . **Er. Subit Raj Pokhrel** Khalti Pvt. Ltd
Signature of HOD . . . . **Mr. Sushant Paudel** Bhaktapur Multiple Campus	**Signature of External Examiner** . . . . **.....** IOST, Tribhuvan University

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this internship report. Special gratitude to my supervisor **Mr. Pratik Timilsina** for the complete support and guidance throughout the internship period.

Also, I would like to express my special gratitude to our Program Coordinator, **Mr. Sushant Paudel** and administrative staff whose all-time encouragement helped me coordinate the internship tasks systematically.

I would like to express my sincere thanks to my mentors **Er. Subit Raj Pokhrel** and **Er. Ashin Shakya** of Khalti Pvt. Ltd. for sharing their valuable knowledge and guiding me during the internship period, and making me learn new skills and abilities. I am also grateful to the entire staff of Khalti Pvt. Ltd for their constant support and guidance

With all due respect and gratitude, I would like to give a word of thanks to the members of the IT department of Bhaktapur Multiple Campus, who encouraged me to perform work activities.

With Regards,
Anukul Adhikari (23226/76)

ABSTRACT

This report examines the crucial role of Backend Python developer intern at Khalti Pvt. Ltd. Known for its versatility, Python is essential for Khalti's projects. Interns are vital to the company's goal of fostering innovation and talent. Through hands-on work, mentorship, and real-world project exposure, they contribute to solutions and enhance their skills. The documented project was assigned to me, and I used popular development tools and techniques within company policy to develop the software. This report covers the duties, challenges, and learning experiences of interns, underlining internships as key for talent acquisition and growth. It also details Khalti's recruitment, onboarding, and mentoring processes, showing the company's dedication to learning and innovation. By providing insights into the role of interns, this report highlights Khalti's commitment to nurturing talent and achieving excellence in software development.

Keywords: *Python, Mentorship, Talent Acquisition, Hands-on Experience,Opportunities*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CI/CD	Continuous Integration and Continuous Development
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
API	Application Programming Interface

Chapter 1: Introduction

1.1. Introduction

Over the course of my Backend Python Developer Internship at Khalti Digital Wallet, the chance was given to fully embrace the software development culture and observe how it has been used effectively to deliver software to the masses. Khalti Digital Wallet is one of the well-known names in innovative financial solutions, offering a wide array of innovative and advanced solutions to commercial banks, development banks, financial institutions, and several other large enterprises. The company is committed to using advanced technology with the view of enhancing financial inclusion and making digital experience seamless.The primary responsibility, as a member of the backend team, has been to learn how to streamline and enhance the development and operational processes, so that the resulting systems could be robust, efficient, and reliable.

The main duty included understanding and implementation of software practices, simplify different parts of the software development process. This entailed creation as well as optimization of various software systems while increasing overall team's productivity.

Additionally, there was an opportunity given to observe how consumers utilize various features and services of a large digital wallet platform. The users were facing difficulty and also in the customer facing department there was lots of issue regarding the email address and verification, thus i worked on the email change/verification system feature.

Real-world experience with Software Development, complementing the theoretical knowledge I previously acquired, has been one of the best outcomes of this internship. A better understanding of the collaborative and iterative nature of software development was also achieved. Khalti has been able to produce quality software faster by focusing on continuous improvement, optimization, and ensuring effective communication between development and operations teams. These skills and ideas will help in handling future challenges in this area, thereby contributing to becoming an even better software engineer.

1.2. Problem Statement

In the development of a digital wallet application at a fintech company, a robust and secure email verification system was required to enhance user account security and ensure the validity of email addresses used for communication. The system needed to address two primary scenarios: the verification of a new user's email address upon signup and the updating of an already verified email address. In both scenarios, it was crucial to ensure that the email verification process was secure, reliable, and user-friendly, preventing unauthorized access and ensuring that only verified email addresses were used within the system.

To achieve this, a feature was developed using Django and Django REST framework for the backend, with Celery employed to send emails asynchronously. The feature involved storing the user's email in the database with an is_email_verified flag set to false upon signup and sending a one-time link to the user's email for verification. For users updating their verified email addresses, a two-step verification process was implemented, where a one-time link was first sent to the old email address and, upon clicking, another one-time link was sent to the new email address. The system ensured that both links were clicked for the email change to be completed and verified. Extensive testing, including unit and integration tests, was conducted to ensure the implementation's effectiveness and reliability.

Addressing these problems was crucial for maintaining Khalti's competitive edge, ensuring customer satisfaction, and supporting the company's growth objectives.

1.3. Objectives

The primary objectives of my internship at Khalti Digltal Wallet were:

i) **To work**:
Work in a real-world corporate environment to understand team dynamics, project management, and effective communication within a professional setting.

ii) **To tackle**:
Tackle real-world challenges and develop solutions, enhancing critical thinking and problem-solving abilities.

iii) **To implement**:
Implement various steps of software development life cycle, to gain hands-on experience developing scalable backend application.

iv) **Collaboration with Senior Developers**:
To collaborate with senior developers and learn best practices in coding and project management.

1.4. Scope and Limitation

1.4.1. Scope

The scope of my internship included the following key areas:

i) **Backend Development Contribution**:
Django and Django REST frameworks used to construct APIs for the digital wallet, including a payment scheduling feature with functionalities for scheduling, rescheduling, and automatic execution.

ii) **Agile Development Participation**:
Involvement in the software development lifecycle, potentially encompassing sprints, code reviews, and collaboration with mentors and teams to deliver the payment scheduling feature.

iii) **Mentorship and Knowledge Acquisition**:
Guidance from experienced developers, understanding of secure backend development for financial applications and best practices within the software development lifecycle.

iv) **Technical Skill Development**:
Enhancement of python programming skills and expertise in the Django framework. Expanding knowledge of financial regulations while addressing challenges and proposing codebase improvements.

1.4.2. Limitations

Despite the comprehensive scope, there were some limitations during my internship:

i) **Limited Project Scope**:
The internship focused primarily on the development of a single feature (email change/verify) within the digital wallet, potentially limiting exposure to the broader functionalities of the application.

ii) **Intern-Level Responsibilities**:
Tasks were assigned with an intern's experience level in mind, potentially offering less opportunity to work on complex backend functionalities or core system architecture.

iii) **Focus on Specific Technologies**:
The internship primarily used Django and Django REST framework, potentially limiting exposure to a wider range of technologies and frameworks used in the industry.

iv) **Time Constraints**:
The internship's timeframe may have restricted the ability to delve deeper into specific technical aspects or explore alternative approaches to the payment scheduling feature.

1.5. Report Organization

This report is structured into four main chapters, each detailing different aspects of my internship experience at Khalti Digital Wallet. Here is a brief overview of each chapter:

i) **Chapter 1: Introduction**
This chapter introduces the work completed during my internship. It outlines the problem statement, the objectives of the internship, the scope and limitations of the project, and provides an overview of the report's organization.

ii) **Chapter 2: Organization Details and Literature Review**
In this chapter, a comprehensive introduction to Khalti Digital Wallet has been provided. This includes an overview of the organization, its hierarchy, the various domains in which it operates,

and a detailed description of the department where internship has been completed. Additionally, this chapter includes a literature review or related study, highlighting relevant theories and frameworks that underpin the works that have been performed during the internship.

iii) **Chapter 3: Internship Activities**

This chapter delves into the specifics of my internship activities. It outlines my roles and responsibilities, provides a weekly log of the technical activities, describes the involved projects, and details the technical tasks and activities have been completed successfully. This section offers an in-depth look at the hands-on experience obtained.

iv) **Chapter 4: Conclusion and Learning Outcomes**

A brief overview of the experience gained during the internship is also stated in this last part, as well as the main conclusions. It mentions my skills and knowledge, challenges I faced and how I dealt with them. Additionally, the section talks about what the future holds in terms of career development after such an opportunity.

Chapter 2: Background Study and Literature Review

2.1. Introduction to Organization

Khalti Digital Wallet is a leading financial technology company based in Nepal. Established in 2017, the company specializes in providing innovative digital financial solutions to a diverse range of clients, including BFIs(Banks, Cooperatives, Securities) and enterprises in Nepal. Khalti's mission is to revolutionize the financial services industry by leveraging cutting-edge technology to enhance financial inclusion and provide seamless digital experiences to its users. The company's portfolio includes a wide array of products and services such as wallet itself, payment gateway, and resellable API services, all designed to meet the evolving needs of the Nepal's modern financial ecosystem. Khalti is recognized for youth centric mobile digital wallet in the country.

The digital wallet developed by Khalti is currently used by over 40 lakhs+ users. The systems contribute to nearly 30% of the total digital wallet users in Nepal. The company's efforts in fintech innovation have been duly recognized by various national and international bodies, including the 2022 WITSA Global Innovation & Tech Excellence Awards. Khalti is also committed to financial literacy and has partnered with The Asia Foundation to provide digital and financial information to Nepali migrant workers through the Shuvayatra app. Funded by the Department for International Development (DFID), this initiative aims to equip migrant workers with the skills they need to make informed financial decisions.

Khalti is the leading player in the fintech industry due to its unwavering dedication to perfection, constant forward thinking and focus on the user base. Financial inclusion for millions is being championed by Khalti with its strong infrastructure and smart team that yearns to see this achieved through providing safe trustworthy easy to use financial solutions.

Table 2.1 : Company Details

Official name	Khalti Digital Wallet
Type of business	Fintech
Location	Bakhundole, lalitpur
Year of establishment	2017
Key service areas	Digital Payment Solutions
Staff size	300
Location of clients	Nepal

Expertise in	Financial Software Development, Digital Payment Systems
Noteworthy mentions	Innovation in Digital Payments, Core Banking Solutions

2.2. Organizational Hierarchy

Khalti Digital Wallet prioritizes innovation through its structure. The Board sets direction, and Executive Management implements it. Specialized departments handle tasks: Product Development builds software, Sales & Marketing drives growth, Customer Support keeps users happy, DevOps streamlines processes, and Finance & Admin manages the company's well-being. This structure fosters collaboration towards Khalti's goals.

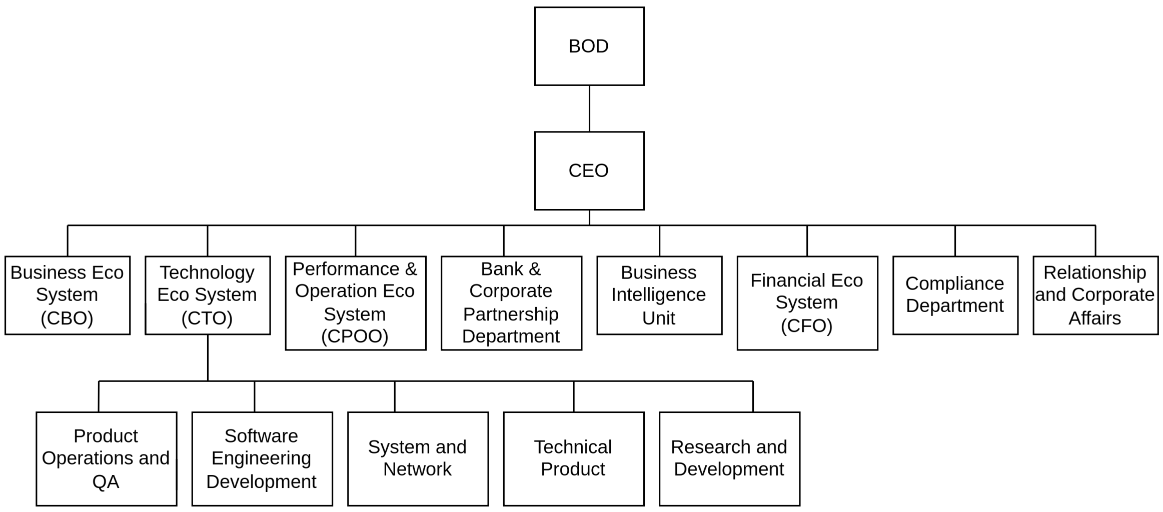

BOD
CEO
Business Eco System (CBO)
Technology Eco System (CTO)
Performance & Operation Eco System (CPOO)
Bank & Corporate Partnership Department
Business Intelligence Unit
Financial Eco System (CFO)
Compliance Department
Relationship and Corporate Affairs
Product Operations and QA
Software Engineering Development
System and Network
Technical Product
Research and Development

Figure 2.1 : Organizational Hierarchy

2.3. Working Domains of Organization

Khalti Pvt. Ltd is among the top technology firms located in Nepal that mainly focuses on financial technology solutions (fintech). Since its establishment in 2017, Khalti has revolutionized digital financial services thus changing Nepalese and regional financial sector landscape. The company operates principally in:

i) **Digital Wallet** - Khalti is popularly known for its digital payment wallet that offers various utilities including Topup, Bill payments, P2P, P2M fund transfers.

ii) **API Aggregation Layer** – The aggregation layers allows for diferent APIs from telecommunications and BFIs for other institutions to leverage without the infrastructure cost.

iii) **Payment Gateway** – Payment gateway allows for merchants to recieve transactions on the go from various differnt channels including PSO(ConnectIPS, Fonepay), mobile banking, internet banking, cards etc.

iv) **Bank Middleware** – The bank middleware allows for integration of core banking systems with the digital wallet and payment gateway for seamless transactions.

2.4. Description of Intern Department

2.4.1 Placement

The Backend Development team plays a critical role in building and maintaining the core functionalities of the Khalti digital wallet application. The team is responsible for developing robust and scalable APIs using frameworks like Django and Django REST to deliver a seamless user experience. This involves designing, implementing, and testing backend logic, ensuring data integrity, and collaborating with other teams to deliver features on schedule. as a Backend Python Developer intern, had the opportunity to work under the guidance of experienced developers at Khalti. This internship provided valuable hands-on experience in building and maintaining a secure backend for a financial application. Responsibilities included:

i) Contributing to the development of the Khalti digital wallet using Django and Django REST framework.

ii) Designing and implementing APIs for functionalities like add/edit email feature.

iii) Collaborating with other developers to ensure code quality, maintainability, and adherence to best practices.

iv) Writing unit tests to ensure the reliability and robustness of the backend code.

This internship at Khalti fostered my technical skills in Python programming and Django framework expertise. It also provided valuable insight into the software development lifecycle within a fast-paced fintech environment.

2.4.2 Duration

The duration and relevant details of internship are as follows:

Table 2.2 : Duration of Internship

Start Date	April 08, 2024
End Date	June 07, 2024
Duration	2 months
Office hours	9:30am −6:30pm

2.5. Literature Review

Python and Django development have seen significant growth in recent years, with a plethora of resources available to developers looking to enhance their skills and stay abreast of industry trends. A review of the literature reveals several key areas of focus, including foundational Python programming concepts, advanced Django development techniques, best practices in web development, and emerging trends in the Python ecosystem.

Foundational Python programming concepts was explored through the official Python documentation, supplemented by books such as (Matthes, 2019) and (Ramalho, 2015), which offer comprehensive coverage of language fundamentals and advanced features.For Django development, the official Django documentation served as an indispensable guide, complemented by books like (Vincent, 2019a) and (Vincent, 2019b), which offered practical insights and real-world examples. Online resources such as Real Python and Simple helped more than Complex provided tutorials, articles, and community forums for Django developers to learn and share best practices.

REST framework and API development are crucial aspects of modern web development. Django REST framework is a powerful and flexible toolkit for building Web APIs in Django. The official DRF documentation, along with books like (Hillar, 2018) and (Vincent, n.d.) , provides detailed explanations and practical examples for creating robust APIs. These resources cover essential topics such as serialization, authentication, permissions, viewsets, and routers. Online tutorials and community forums also play a vital role in helping developers understand and implement RESTful APIs, ensuring they follow best practices and keep up with industry standards.

Token-based authentication methods have become a cornerstone for securing web applications and APIs, offering a scalable and stateless mechanism for user authentication. In Django and Django REST Framework, several token-based authentication methods are employed, each providing distinct advantages and fitting different use cases.Token Authentication is a method integrated within DRF, leveraging a simplistic approach where the server issues a token upon successful user login. Each subsequent request must include this token for user authentication. This method is well-documented in the official DRF documentation. According to the documentation, a significant advantage of Token Authentication is its straightforward implementation, making it suitable for small-scale applications. However, a notable limitation is the lack of built-in token expiration, which can pose security risks if tokens are compromised.

Chapter 3: Internship Activities

3.1. Roles and Responsibilities

While working as a Backend Python Developer intern for Khalti, my main focus was in the development and refinement of a various feature, which involved designing and implementing backend solutions using Python and Django, ensuring these systems were robust, scalable, and seamlessly integrated with the existing digital wallet platform.I had the following tasks:

i) **Code Development and Testing**:
Worked on new backend features and enhanced existing functionalities using Python and Django, and I conducted thorough testing to ensure robustness and reliability.

ii) **System Maintenance and Optimization**:
Worked out routine maintenance and optimization tasks to improve the efficiency and scalability of the backend systems.

iii) **Database Management**:
I managed database schemas and integrated data storage solutions, ensuring data consistency and security.

iv) **Implementation of Security Protocols**:
Implemented and reviewed security measures to safeguard the digital wallet against potential threats and vulnerabilities.

v) **Collaboration with Other Teams**:
Maintained close collaboration with the front-end development team and other technical departments to ensure seamless integration and functionality across platforms.

vi) **Documentation of Development Processes**:
Created and maintained comprehensive documentation of all development processes, changes, and upgrades, providing a clear reference for future development efforts and troubleshooting.

3.2. Weekly log

Following table shows the weekly activities the intern performed throughout their internship period.

Table 3.1 : Weekly Log

Week	Activities
Week 1	• Introduced to the team and company culture. • Setup of personal development environment and necessary software. • Overview of the digital wallet codebase with a senior developer.

	• Attended workshops on Python and Django, the core technologies used.
Week 2	• Studied existing email functionalities within the digital wallet. • Reviewed documentation of related backend processes. • Participated in daily stand-ups to discuss ongoing projects. • Learned about security protocols relevant to user data handling.
Week 3	• Began coding the email update feature in the user profile section. • Implemented backend logic for sending verification emails. • Set up database migrations to handle new email data securely. • Developed error handling routines for failed email updates.
Week 4	• Enhanced error handling routines for failed email updates. • Improved user interface for email update feature. • Coordinated with frontend developers to align on feature integration. • Drafted initial user guides for the new feature.
Week 5	• Integrated the new email functionality with the frontend interface. • Conducted preliminary testing with dummy data to ensure stability. • Identified bugs and issues in the initial deployment. • Refined the API responses for better user feedback on errors.
Week 6	• Submitted code for peer review and incorporated feedback. • Optimized database queries for faster email updates. • Enhanced security measures based on latest best practices. • Wrote comprehensive unit tests for the new features.
Week 7	• Deployed the feature in a controlled test environment. • Monitored user interactions and collected feedback from test users. • Updated project documentation to include new features and changes. • Prepared a rollback plan for potential deployment issues.
Week 8	• Made final adjustments based on user feedback and testing results. • Conducted a final review with the project team and stakeholders. • Officially launched the email change feature to production. • Completed a detailed handover to the maintenance team, ensuring they are equipped to manage and troubleshoot the feature.
Week 9	• Analyzed post-launch metrics to measure feature adoption and performance.

	• Addressed any critical issues reported by users. • Continued optimizing the feature for performance and security. • Began documenting lessons learned from the project.
Week 10	• Participated in a retrospective meeting to discuss project successes and areas for improvement. • Provided training sessions for the support team on the new feature. • Assisted in resolving any outstanding user issues. • Worked on minor feature enhancements based on user feedback.
Week 11	• Collaborated with marketing to create promotional materials for the new feature. • Monitored long-term performance and user satisfaction. • Identified potential areas for future improvements. • Began planning for the next feature development cycle.
Week 12	• Completed a comprehensive project report summarizing all activities and outcomes. • Held a final project wrap-up meeting with all stakeholders. • Provided final updates to the project documentation. • Celebrated the successful completion of the project with the team.

3.3. Description of the Project(s) Involved During Internship

One of the highlights which really helped me in understanding whole software development process was working on two minor projects during my internship both using django and django rest framework.

3.3.1. Allow users to securely edit/change email address

In this project, my main goal was to streamline the process of editing/verifying email address focusing on ease of use and security in mind.

i) The project is done using Django, Django Rest Framework, Django Templates, and Celery.
ii) The project involved creating a new API endpoint for users to change their email address.
iii) The project also involved creating a Celery task to send an email verification link to the new email address.
iv) The system was designed to ensure that the email change process was secure and user-friendly.

3.3.1.1 Functional Requirement

The functional requirements for a system describe what the system should do. Those requirements depend on the kind of software being developed and the expected software users. These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in specific situations.

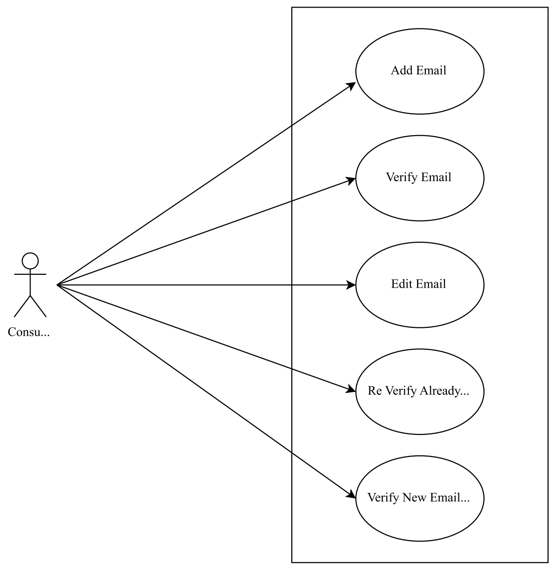


Figure 3.1 : Use case diagram

The Figure 3.1 represents a consumer and system interaction that shows the basic working features of the application itself, where the user can change their email address and verify it.

Use Case Description

Use Case ID	User Case Name	Actor	Description	Preconditions	Main Flow	Alternative Flow	Postconditions
UC01	View Profile	Consumer	Users can view additional options in their profile.	User is logged into the Khalti App and is in the Profile section.	i) User accesses profile.2. User views 3 dots for more options.	None	User sees 3 dots.
UC02	Add Email	Consumer	Allows the user to add an email ad-	User is in the Profile section of	i) User clicks on 3 dots.2.	i) Invalid email format.2.	Email is added pending

Use Case ID	User Case Name	Actor	Description	Preconditions	Main Flow	Alternative Flow	Postconditions
			dress to their profile.	the Khalti App.	User selects “Add Email”.3. User enters email and submits for authentication.	Email linked to another account.	verification.
UC03	Authenticate Email Addition	Consumer	Handles authentication required for adding an email.	User has entered a valid email address.	i) User is prompted for authentication.2. User completes authentication.	i) Biometric fails, try password.2. Incorrect password input.	Email is authenticated, verification link sent.
UC04	Update Email	Consumer	Allows the user to change their email address.	User has previously added an email.	i) User accesses “Update Email” from profile.2. User enters new	Same as Add Email use case.	Email update process initiated.

Use Case ID	User Case Name	Actor	Description	Preconditions	Main Flow	Alternative Flow	Postconditions
					email and authenticates.		
UC05	Handle Authentication Failures	Consumer	Manages repeated authentication failures.	User has failed authentication attempts.	User attempts to authenticate and fails.	User fails multiple times leading to lock-out.	User is temporarily locked out after specified failures.
UC06	Handle Expired Verification Link	Consumer	Manages scenarios where a user clicks an expired verification link.	User has received a verification link.	User clicks expired verification link.	None	User is directed to a webpage to request a new link.
UC07	Resend Verification Email	Consumer	Allows users to resend the verification email if not received.	User has not received the initial verification email.	User selects the option to resend verification email.	None	Verification email is resent.
UC08	Handle Network Issues During Email Submission	Consumer	Manages scenarios with network connectivity issues during email submission.	User is adding an email.	User faces network issues during the submission.	None	Error message is displayed, advises checking network.

3.3.1.2 Non-Functional Requirement

Non-functional requirements are requirements that are not directly concerned with the specified function delivered by the system.

- **Maintainability**:
  Adhere to coding standards that promote readability and maintainability.
- **Compliance**:
  Ensure the system complies with relevant data protection regulations (like GDPR, if applicable) regarding user data.

3.3.1.3 Feasibility Study

Before starting the project, a feasibility study is carried out to measure the system's viability. A feasibility study is necessary to determine if creating a new or improved system is friendly with the cost, benefits, operation, technology, and time. The following are the feasibility concerns in this project:

i) **Technical Feasibility**:

The technical feasibility of enhancing the email edit/update feature in a Django-based system considers both the current technological framework and specific security requirements essential in the financial sector. Given that Django is robust in managing secure user interactions and database modifications (necessary for updating email addresses), the existing infrastructure is likely sufficient. However, due to the sensitive nature of financial data, the security implementations need to be rigorously evaluated. This includes ensuring encryption for data transmissions and safeguarding against vulnerabilities specific to financial applications, such as phishing and session hijacking.

i) **Operational Feasibility**:
From an operational feasibility standpoint, integrating this feature into Khalti's digital wallet platform is seamless and minimally disruptive to existing operations. Since digital wallets require high reliability and user trust, any changes that involve user account details, like email addresses, must be handled with extreme caution to avoid eroding trust or introducing errors. The introduction of the feature should be accompanied by comprehensive user documentation and possibly a brief tutorial within the app to facilitate adoption. Additionally, since this feature is fundamental to account security, ensure that your customer support team is well-prepared to address any issues that arise quickly and effectively. This might involve specialized training and updating internal operational protocols.

ii) **Economic Feasibility**:
The economic feasibility for Khalti involves a detailed analysis of costs versus anticipated benefits. The direct costs include development, testing, additional security measures, and training. Operationally, the feature should lead to a reduction in support costs, as users can self-manage their email details, potentially lowering the volume of support requests related to account access

issues. From a benefits perspective, enhancing user autonomy and security can improve customer satisfaction and retention—a crucial metric in the competitive fintech space. Calculating the return on investment should factor in these indirect benefits, such as enhanced user trust and reduced risk of security breaches, which can have significant financial implications. As a fintech entity, projecting the feature's impact on enhancing regulatory compliance and reducing fraud incidents could further justify the investment.

3.3.1.4 System Design

System design defines the components, modules, interfaces, and data for a system to satisfy specified requirements. It can also be defined as creating or altering systems and the processes, practices, models, and methodologies used to develop them. The main objective of the detailed system design is to prepare a system blueprint that meets the goals of the conceptual system design requirements. The system designs for building this project include database schema, input-output design, class diagram, sequence diagram, and activity diagram.

3.3.1.4 Architectural Design

The architectural design shows the architecture of the overall system. The system is based on MVC architecture.

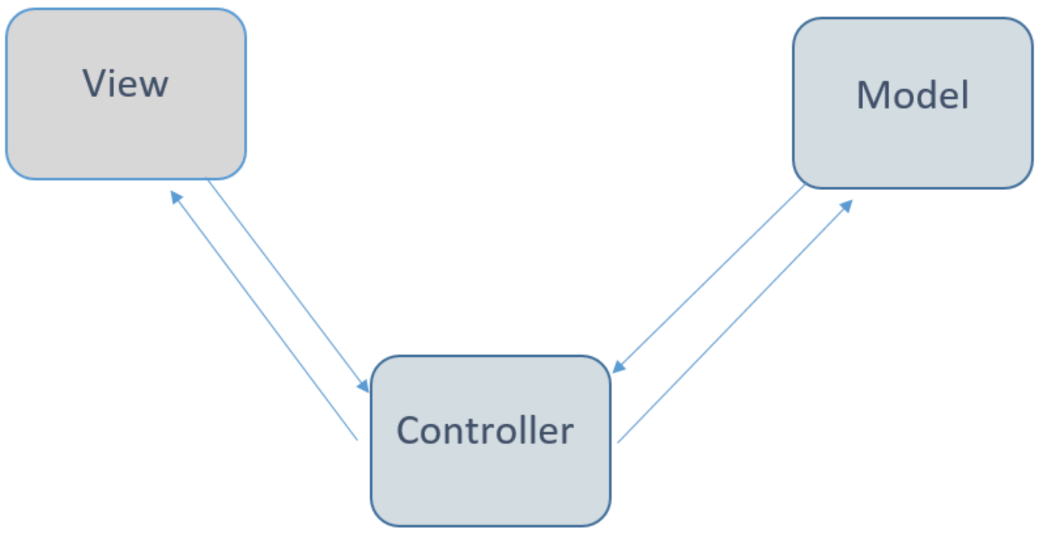


Figure 3.2 : Architectural Design of the System

3.3.1.5 Database Design

The following database schema is the general structure used in the application.

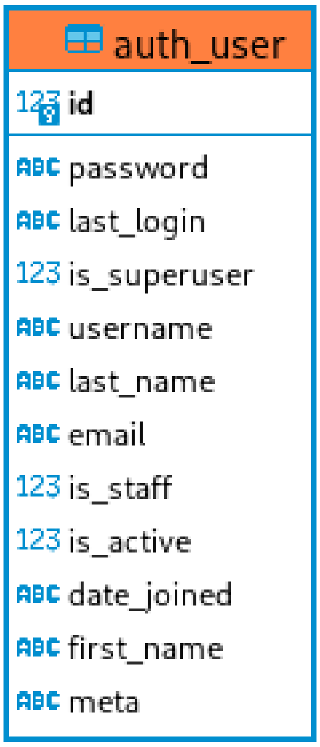


Figure 3.3 : Table of User Model

3.3.1.6 Class Diagram

The class diagram describes the relationships and source code dependencies among other classes.

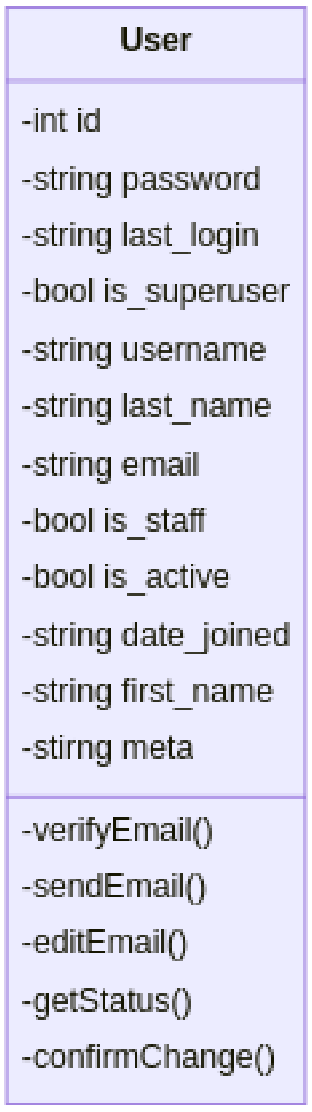


Figure 3.4 : Class Diagram of User Model

The Figure 3.3 illustrates the class diagram of the system. There is only one class, User, which is responsible for handling user data and operations on itself.

3.3.1.7 Sequence Diagram

A sequence diagram is an interaction diagram because it describes how and in what order a group of objects works together. Sequence diagrams are sometimes known as event diagrams. The sequence diagram of the system is shown below.

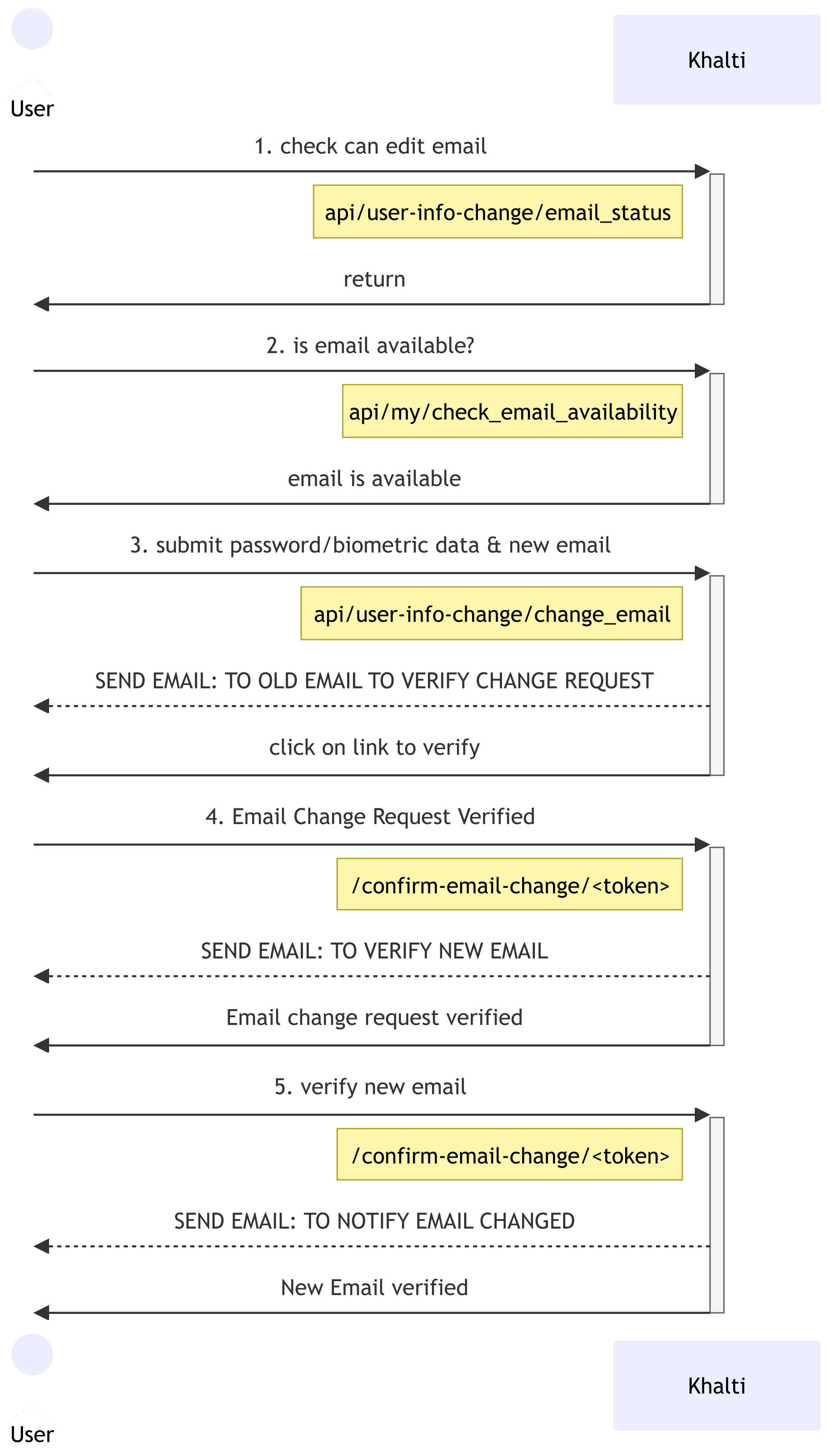

Khalti
User
1. check can edit email
api/user-info-change/email_status
return
2. is email available?
api/my/check_email_availability
email is available
3. submit password/biometric data & new email
api/user-info-change/change_email
SEND EMAIL: TO OLD EMAIL TO VERIFY CHANGE REQUEST
click on link to verify
4. Email Change Request Verified
/confirm-email-change/<token>
SEND EMAIL: TO VERIFY NEW EMAIL
Email change request verified
5. verify new email
/confirm-email-change/<token>
SEND EMAIL: TO NOTIFY EMAIL CHANGED
New Email verified
User
Khalti

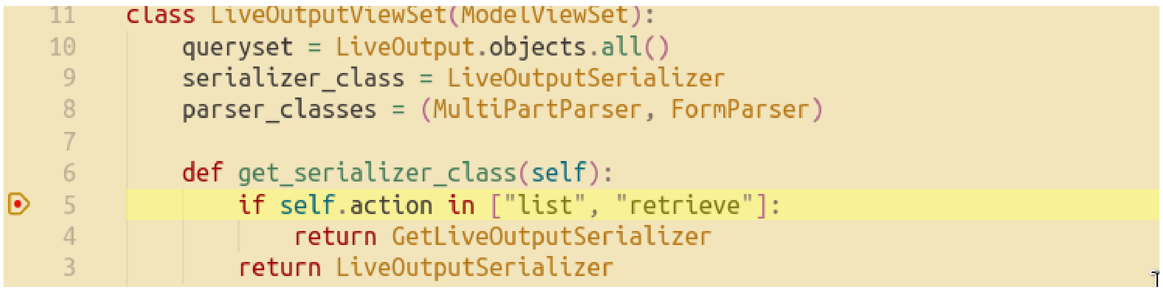
Figure 3.5 : Sequence Diagram of Email change

The Figure 3.3 illustrates how the frontend system will interact with the backend api to change the email address of the user.

3.4. Tasks / Activities Performed

i) Python Debugger

The utilization of the VS Code debugger significantly enhanced my development workflow by allowing me to efficiently identify and resolve issues within the codebase. This tool provided an interactive environment where I could set breakpoints, inspect variables, and step through the code line by line. Consequently, it helped me understand the flow of the application and pinpoint the exact locations of bugs. This experience not only increased my debugging efficiency but also deepened my comprehension of the underlying code structure, leading to more robust and error-free implementations.

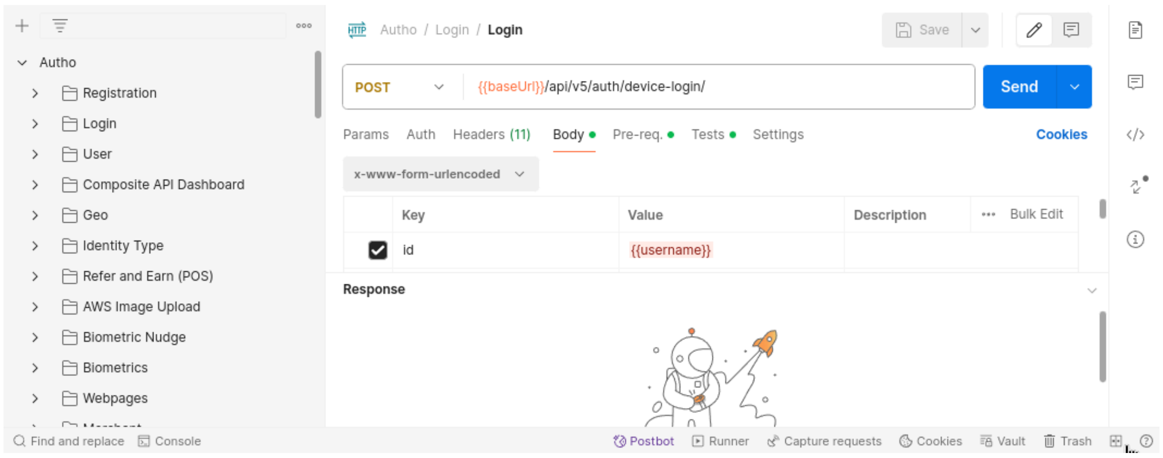


```python
class LiveOutputViewSet(ModelViewSet):
    queryset = LiveOutput.objects.all()
    serializer_class = LiveOutputSerializer
    parser_classes = (MultiPartParser, FormParser)

    def get_serializer_class(self):
        if self.action in ["list", "retrieve"]:
            return GetLiveOutputSerializer
        return LiveOutputSerializer
```

Figure 3.6 : Breakpoint in VS Code Debugger

ii) Using Postman

Using Postman was instrumental in testing and verifying the APIs I developed. This tool enabled me to create, send, and manage HTTP requests, facilitating thorough testing of endpoints and ensuring they behaved as expected. By simulating various scenarios and examining the responses, I was able to identify and rectify issues early in the development process. This practice improved the reliability and performance of the APIs, ensuring seamless integration with other system components and delivering a better user experience.



Autho
- Registration
- Login
- User
- Composite API Dashboard
- Geo
- Identity Type
- Refer and Earn (POS)
- AWS Image Upload
- Biometric Nudge
- Biometrics
- Webpages

Autho / Login / **Login**

POST {{baseUrl}}/api/v5/auth/device-login/ **Send**

Params Auth Headers (11) **Body** Pre-req. Tests Settings Cookies

x-www-form-urlencoded

	Key	Value	Description
✓	id	{{username}}	

Response

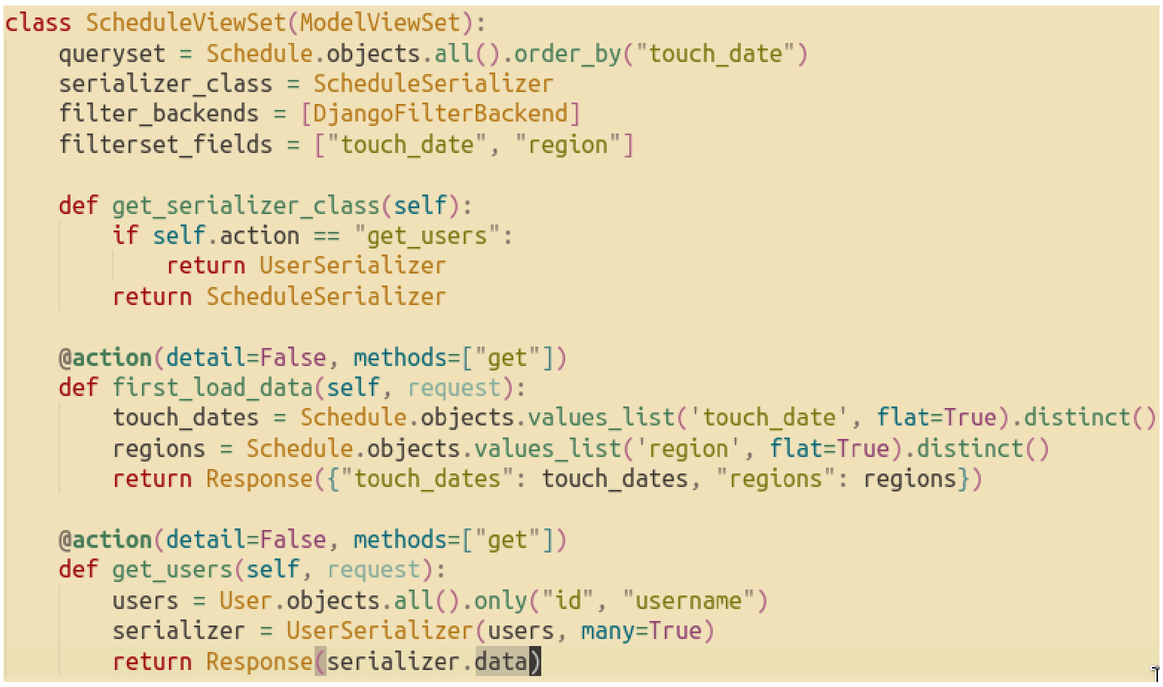
Figure 3.7 : API collections in Postman

iii) Using Git (Merge, Fetch, Cherry-pick)

Mastering Git commands such as merge, fetch, and cherry-pick significantly streamlined my version control processes. Git merge allowed me to combine multiple branches into a cohesive codebase, ensuring all features and fixes were integrated smoothly. Git fetch helped keep my local repository up to date with the remote repository, preventing discrepancies and potential conflicts. Git cherry-pick was particularly useful for selectively applying specific commits from one branch to another, facilitating precise and controlled code updates. These skills enhanced my ability to manage code changes effectively, maintain a clean project history, and collaborate seamlessly with team members.

iv) In-Depth Understanding of DRF (Django Rest Framework)

Developing an in-depth understanding of how the Django Rest Framework (DRF) works, including viewsets and routers, was pivotal in building scalable and maintainable APIs. By comprehending the intricacies of viewsets, I could encapsulate common patterns for interacting with the data in a reusable manner. Understanding routers enabled me to automate URL routing, reducing boilerplate code and potential errors. This knowledge empowered me to construct more efficient and organized APIs, facilitating easier maintenance and extensibility of the codebase.

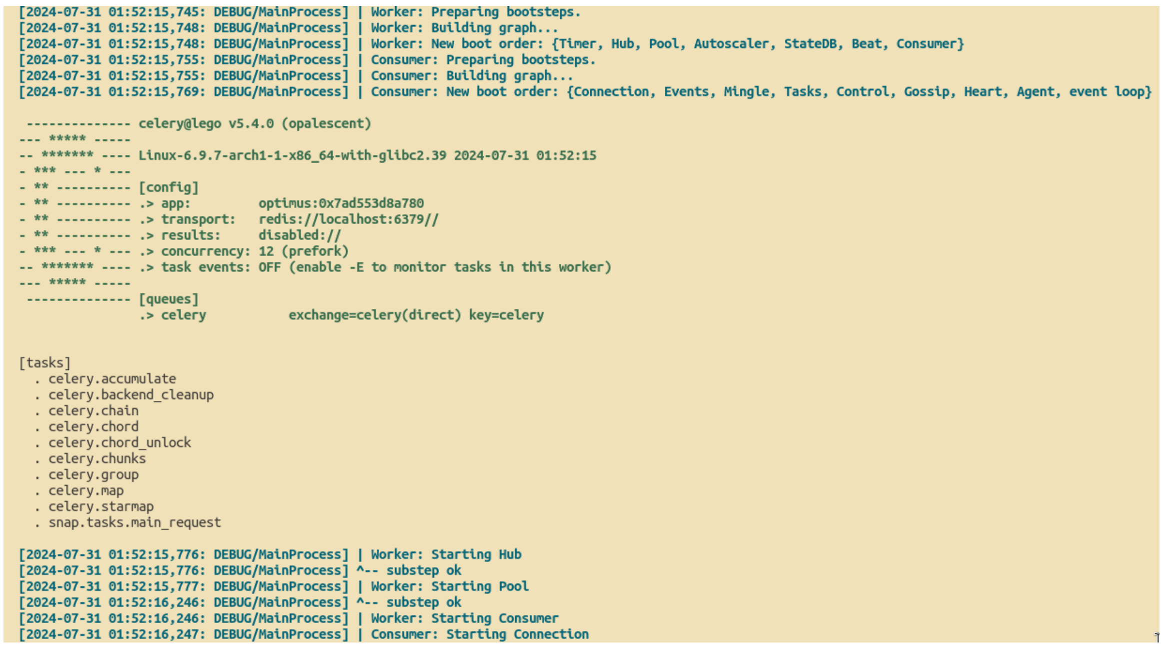


```python
class ScheduleViewSet(ModelViewSet):
    queryset = Schedule.objects.all().order_by("touch_date")
    serializer_class = ScheduleSerializer
    filter_backends = [DjangoFilterBackend]
    filterset_fields = ["touch_date", "region"]

    def get_serializer_class(self):
        if self.action == "get_users":
            return UserSerializer
        return ScheduleSerializer

    @action(detail=False, methods=["get"])
    def first_load_data(self, request):
        touch_dates = Schedule.objects.values_list('touch_date', flat=True).distinct()
        regions = Schedule.objects.values_list('region', flat=True).distinct()
        return Response({"touch_dates": touch_dates, "regions": regions})

    @action(detail=False, methods=["get"])
    def get_users(self, request):
        users = User.objects.all().only("id", "username")
        serializer = UserSerializer(users, many=True)
        return Response(serializer.data)
```

Figure 3.8 : ModelViewSet of Django Rest Framework

v) Learning About Celery Tasks

Learning about Celery tasks and their implementation in my projects played a crucial role in optimizing performance and handling asynchronous operations. By offloading time-consuming tasks, such as sending emails or processing large datasets, to Celery, I was able to ensure that the main application remained responsive and efficient. This capability not only enhanced the user experience by reducing latency but also improved the scalability of the application, allowing it to handle a higher volume of concurrent operations seamlessly.



```
[2024-07-31 01:52:15,745: DEBUG/MainProcess] | Worker: Preparing bootsteps.
[2024-07-31 01:52:15,748: DEBUG/MainProcess] | Worker: Building graph...
[2024-07-31 01:52:15,748: DEBUG/MainProcess] | Worker: New boot order: {Timer, Hub, Pool, Autoscaler, StateDB, Beat, Consumer}
[2024-07-31 01:52:15,755: DEBUG/MainProcess] | Consumer: Preparing bootsteps.
[2024-07-31 01:52:15,755: DEBUG/MainProcess] | Consumer: Building graph...
[2024-07-31 01:52:15,769: DEBUG/MainProcess] | Consumer: New boot order: {Connection, Events, Mingle, Tasks, Control, Gossip, Heart, Agent, event loop}

 -------------- celery@lego v5.4.0 (opalescent)
--- ***** -----
-- ******* ---- Linux-6.9.7-arch1-1-x86_64-with-glibc2.39 2024-07-31 01:52:15
- *** --- * ---
- ** ---------- [config]
- ** ---------- .> app:         optimus:0x7ad553d8a780
- ** ---------- .> transport:   redis://localhost:6379//
- ** ---------- .> results:     disabled://
- *** --- * --- .> concurrency: 12 (prefork)
-- ******* ---- .> task events: OFF (enable -E to monitor tasks in this worker)
--- ***** -----
 -------------- [queues]
                .> celery           exchange=celery(direct) key=celery


[tasks]
  . celery.accumulate
  . celery.backend_cleanup
  . celery.chain
  . celery.chord
  . celery.chord_unlock
  . celery.chunks
  . celery.group
  . celery.map
  . celery.starmap
  . snap.tasks.main_request

[2024-07-31 01:52:15,776: DEBUG/MainProcess] | Worker: Starting Hub
[2024-07-31 01:52:15,776: DEBUG/MainProcess] ^-- substep ok
[2024-07-31 01:52:15,777: DEBUG/MainProcess] | Worker: Starting Pool
[2024-07-31 01:52:16,246: DEBUG/MainProcess] ^-- substep ok
[2024-07-31 01:52:16,246: DEBUG/MainProcess] | Worker: Starting Consumer
[2024-07-31 01:52:16,247: DEBUG/MainProcess] | Consumer: Starting Connection
```

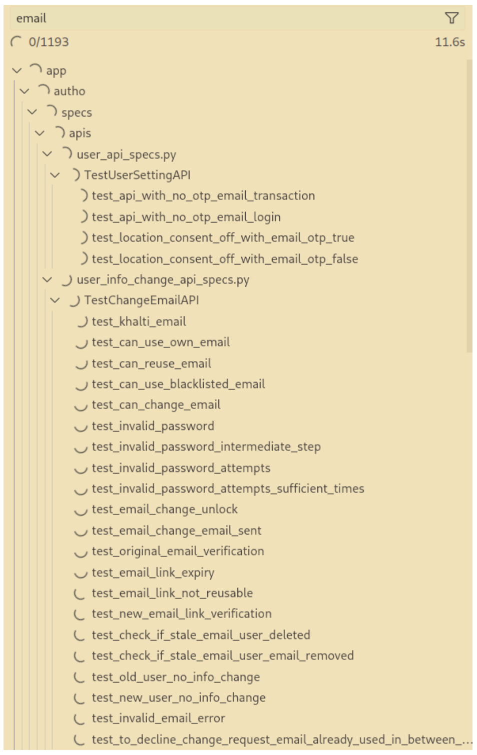
Figure 3.9 : Celery tasks for asynchronous operations

vi) Writing Extendable Code in Python and Advanced OOP Concepts

Learning to write extendable code in Python and mastering advanced object-oriented programming (OOP) concepts significantly improved my coding practices. By adhering to principles such as inheritance, polymorphism, and encapsulation, I was able to create modular and reusable code components. This approach facilitated easier maintenance and scalability of the projects. Understanding design patterns and implementing them appropriately also ensured that the codebase remained flexible and adaptable to future requirements, thereby enhancing the overall software quality and longevity.

vii) Writing Test Cases and Documentation

Developing skills in writing comprehensive test cases and detailed documentation was crucial in ensuring the reliability and maintainability of the code. Leveraging markdown, Mermaid.js to build beautiful readable code with proper usecasediagrams, flow charts and api flows. By covering various test scenarios and edge cases, I could identify and fix potential issues before they reached production, thereby reducing bugs and improving software quality. Writing clear and concise documentation helped create a reliable reference for current and future developers, ensuring that the project could be easily understood, maintained, and extended. This practice promoted better collaboration within the team and facilitated smoother onboarding of new members.



```
email
0/1193                                                    11.6s
app
  autho
    specs
      apis
        user_api_specs.py
          TestUserSettingAPI
            test_api_with_no_otp_email_transaction
            test_api_with_no_otp_email_login
            test_location_consent_off_with_email_otp_true
            test_location_consent_off_with_email_otp_false
        user_info_change_api_specs.py
          TestChangeEmailAPI
            test_khalti_email
            test_can_use_own_email
            test_can_reuse_email
            test_can_use_blacklisted_email
            test_can_change_email
            test_invalid_password
            test_invalid_password_intermediate_step
            test_invalid_password_attempts
            test_invalid_password_attempts_sufficient_times
            test_email_change_unlock
            test_email_change_email_sent
            test_original_email_verification
            test_email_link_expiry
            test_email_link_not_reusable
            test_new_email_link_verification
            test_check_if_stale_email_user_deleted
            test_check_if_stale_email_user_email_removed
            test_old_user_no_info_change
            test_new_user_no_info_change
            test_invalid_email_error
            test_to_decline_change_request_email_already_used_in_between_...
```

Figure 3.10 : Email test cases

Chapter 4: Conclusion and Learning Outcomes

4.1. Conclusion

As wrap up to internship at Khalti Pvt Ltd,looking back on the diverse experiences that have shaped growth as a Python developer and a professional in software development. During this enriching time, I delved deep into Python programming, using its flexibility and power to create strong solutions. From scraping complex data on the web to building RESTful APIs with Django and Django Rest Framework, each task challenged me and helped me improve technically. I focused on writing code that not only worked perfectly but also followed the best practices and standards.

At Khalti Pvt Ltd, teamwork was crucial, regularly discussing our progress in meetings and supported each other. Giving and receiving feedback during code reviews made all better developers. Besides technical skills, I also learned valuable life lessons. I discovered how to persevere through challenges, how open communication enhances team collaboration, and how learning is a continuous journey.

4.2. Learning Outcome

Here are the key areas where I gained substantial knowledge and practical experience:

i) **Technical Skills Development:** Enhanced proficiency in Python programming language, including advanced concepts and libraries,acquired practical experience in web development frameworks such as Django and Django Rest Framework. Mastered techniques for web scraping, API development, database management, and task automation.

ii) **Collaboration and Teamwork:** Developed strong communication and collaboration skills through active participation in team meetings, discussions, and code reviews.Worked effectively in a team environment, contributing ideas, sharing progress, and providing feedback to colleagues.

iii) **Problem-Solving and Critical Thinking:** Discovering and developing analytical and problem-solving skills was a significant aspect of internship journey.Learned how to dig deep to find the root causes of issues and devise effective solutions. Through this process, gained insights into own capabilities and strengths. It was empowering to see myself grow and tackle challenges with greater confidence and competence. This newfound understanding of abilities has been invaluable, shaping my approach to problem-solving and contributing to my personal and professional development.

References

Hillar, G. C. (2018). *Django RESTful Web Services* (3rd ed., Vol. 0). Packt Publishing. https://dokumen.pub/django-restful-web-services-1nbsped-9781788833929.html

Matthes, E. (2019). *Python Crash Course: A Hands-On* (3rd ed., Vol. 0). No Starch Press. https://nostarch.com/python-crash-course-3rd-edition

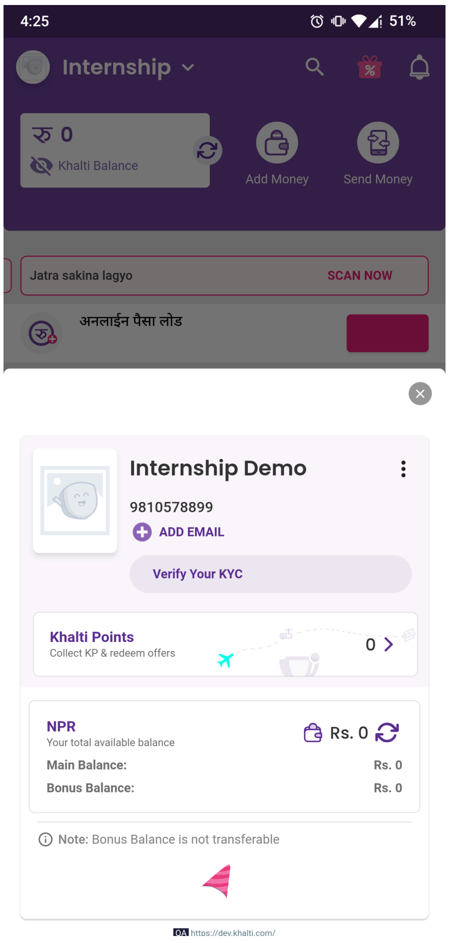
Ramalho, L. (2015). (3rd ed., Vol. 0). O'Reilly Media. https://openlibrary.org/books/OL27112900M/Fluent_Python

Vincent, W. S. *Building APIs with Django and Django Rest Framework*. Independently Published.

Vincent, W. S. (2019a). *Django for Beginners: Build Websites with Python and Django* (3rd ed., Vol. 0). https://books.google.com.np/books?id=GVxwDwAAQBAJ&printsec=frontcover&source=gbs_book_other_versions_r&redir_esc=y#v=onepage&q&f=false

Vincent, W. S. (2019b). *Django for Beginners: Build Websites with Python and Django* (3rd ed., Vol. 0). https://www.amazon.com/Django-APIs-Build-web-Python/dp/1093633948

Appendices



Internship Demo

9810578899

Verify Your KYC

Khalti Points

Collect KP & redeem offers

NPR

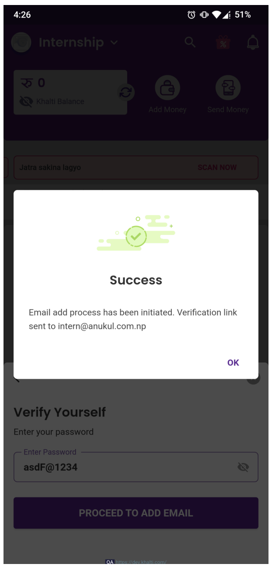
Your total available balance

Main Balance: Rs. 0

Bonus Balance: Rs. 0

Note: Bonus Balance is not transferable

Initializing Locally Hosted Container Registry



Success

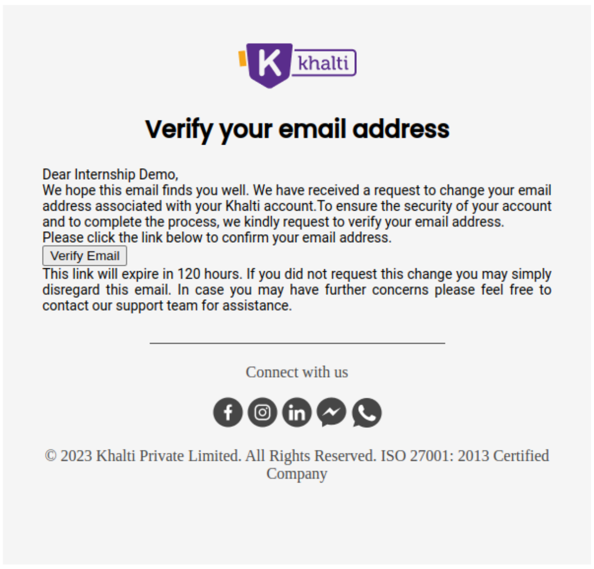
Email add process has been initiated. Verification link sent to intern@anukul.com.np

Verify Yourself

Enter your password

Enter Password: asdF@1234

Mobile popup showing email change process has started



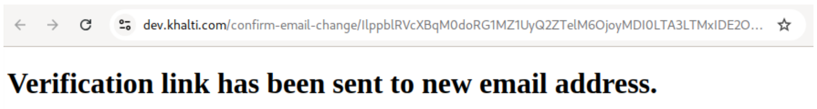
Verify your email address

Dear Internship Demo,
We hope this email finds you well. We have received a request to change your email address associated with your Khalti account.To ensure the security of your account and to complete the process, we kindly request to verify your email address.
Please click the link below to confirm your email address.
Verify Email
This link will expire in 120 hours. If you did not request this change you may simply disregard this email. In case you may have further concerns please feel free to contact our support team for assistance.

Connect with us

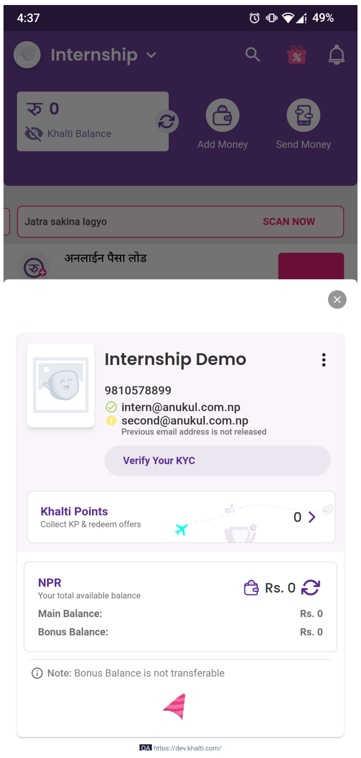
© 2023 Khalti Private Limited. All Rights Reserved. ISO 27001: 2013 Certified Company

Email sent to old email address


dev.khalti.com/confirm-email-change/IlppblRVcXBqM0doRG1MZ1UyQ2ZTelM6OjoyMDI0LTA3LTMxIDE2O...

Verification link has been sent to new email address.

Message shown when verification link clicked

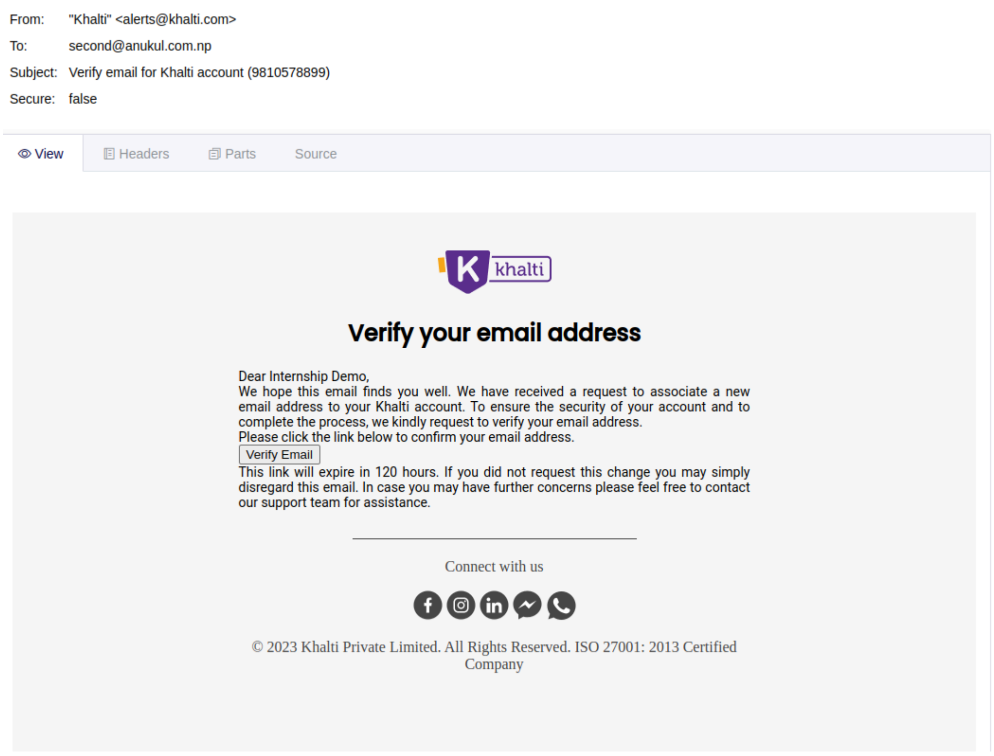


Internship Demo
9810578899
intern@anukul.com.np
second@anukul.com.np
Previous email address is not released

NPR
Your total available balance
Rs. 0
Main Balance: Rs. 0
Bonus Balance: Rs. 0

Note: Bonus Balance is not transferable

Intermediate state of email change



From: "Khalti" <alerts@khalti.com>
To: second@anukul.com.np
Subject: Verify email for Khalti account (9810578899)
Secure: false

Verify your email address

Dear Internship Demo,
We hope this email finds you well. We have received a request to associate a new email address to your Khalti account. To ensure the security of your account and to complete the process, we kindly request to verify your email address.
Please click the link below to confirm your email address.

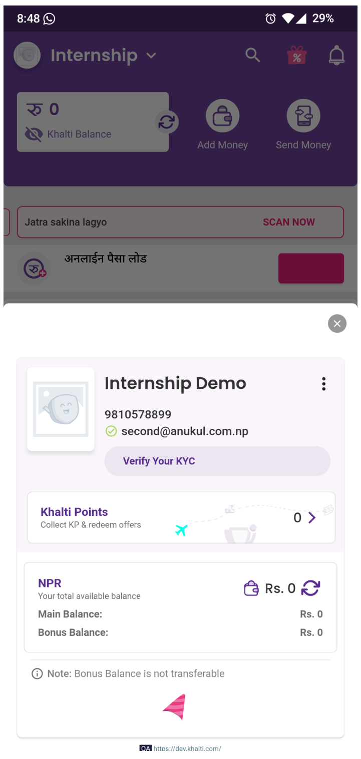
Verify Email

This link will expire in 120 hours. If you did not request this change you may simply disregard this email. In case you may have further concerns please feel free to contact our support team for assistance.

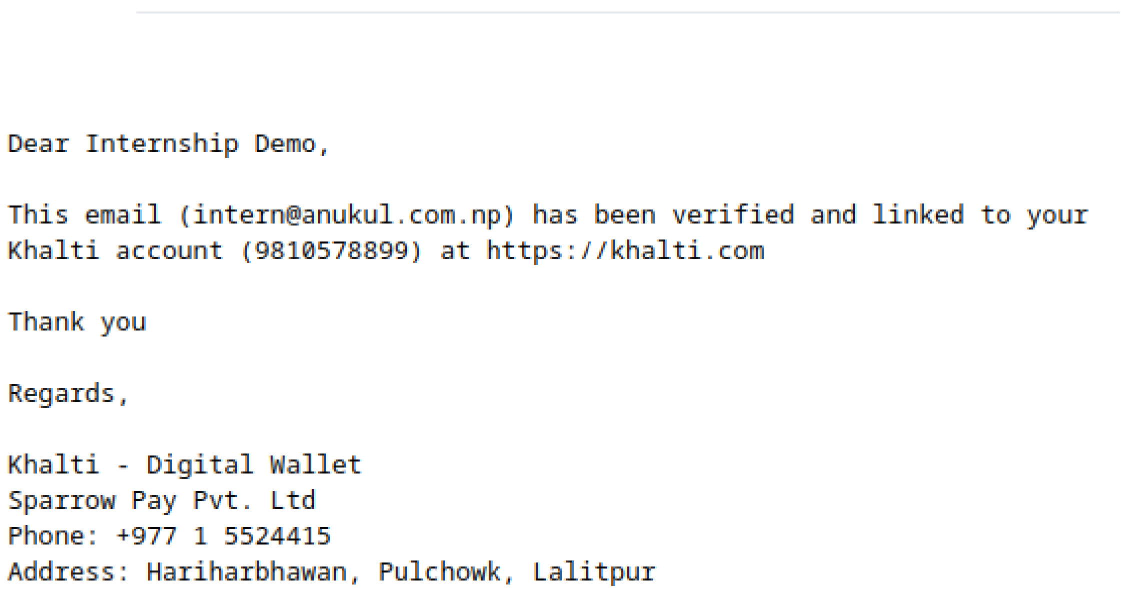
Connect with us

© 2023 Khalti Private Limited. All Rights Reserved. ISO 27001: 2013 Certified Company

Email sent to new address for final verification



Profile UI showing verified email address



```
Dear Internship Demo,

This email (intern@anukul.com.np) has been verified and linked to your
Khalti account (9810578899) at https://khalti.com

Thank you

Regards,

Khalti - Digital Wallet
Sparrow Pay Pvt. Ltd
Phone: +977 1 5524415
Address: Hariharbhawan, Pulchowk, Lalitpur
```

Email sent to newly verified address