



Tribhuvan University

Kathford International College of Engineering and Management

A Final Year Internship Report

On

Revenue Platform

At

SimplySoft Technology Pvt. Ltd.

Submitted To

Department of Information Technology

Kathford International College of Engineering and Management

**In partial fulfillment of the requirements for the Bachelor's degree in Computer
Science and Information Technology (B.Sc. CSIT)**

Submitted By:

Prakriti Bista (15229/074)

September 2022

Date: 22 Bhadra, 2079

TO WHOM IT MAY CONCERN

This is to certify that Ms. Prakriti Bista has successfully completed her internship program as an **Angular Developer** at **Simply Software Technology Pvt. Ltd.** from 27 Chaitra, 2078 to 24 Asadha, 2079.

During this period, Ms. Bista was exposed to various tools and techniques related to the AngularJS framework. She has always displayed a good work ethic and has proven herself to be a hardworking individual. She has been a very good team player and demonstrated a disciplined and professional attitude throughout her tenure. She can work independently and in a team. She is always eager to learn new things and implement them. We are really impressed to see her performance and progress rate on all the tasks and projects that she was assigned to. She has achieved a lot during her short internship period.

We wish her all the best in her future endeavors.




Mr. Kunjan Pandey
Managing Director





Supervisor's Recommendation

I hereby recommend that this report has been prepared under my supervision by Prakriti Bista (15229/074) entitled "Revenue Platform" in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

.....

Mr. Mukesh Prasad Chaudhary

Supervisor/Project Coordinator

Department of Information Technology

Kathford International College of Engineering and Management

Balkumari, Lalitpur

CERTIFICATE OF APPROVAL

This is to certify that this internship report was prepared by Prakriti Bista (15229/074) entitled “Angular Development Internship” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality of a project for the required degree.

<p>.....</p> <p>Mr. Mukesh Prasad Chaudhary</p> <p>Project Supervisor/Internal Examiner Department of IT Kathford International College of Engineering and Management Balkumari, Lalitpur</p>	<p>.....</p> <p>Mr. Bishon Lamichhane</p> <p>Deputy Head of Department Department of IT Kathford International College of Engineering and Management Balkumari, Lalitpur</p>
<p>.....</p> <p>(External Examiner)</p> <p>Tribhuvan University Kirtipur, Nepal</p>	

Acknowledgement

I would like to express my sincere gratitude to SimplySoft Technology Pvt. Ltd for giving me the chance complete my internship in a professional setting, which was a fantastic opportunity for learning and growth. I would like to thank my mentor **Mr. Kunjan Pandey**, Managing Director at Simply Soft, for his advice and support throughout the internship. I am also grateful to Senior Angular Developer Suraj Sir and Roshan Sir, with whom I worked closely who provided me the necessary guidance and showed me the way to achieve my goals. I'm grateful to all the mentors and staff of the organization for their valuable cooperation, guidance, direction, and continuous support throughout the internship period.

I would like to thank **Mr. Mukesh Prasad Chaudhary**, my project supervisor, for his continuous support throughout the course of this internship who provided me with the necessary information and all those important suggestions and corrections he made during the overall internship activities. I would also like to express our special thanks to **Mr. Bishon Lamichhane**, Deputy Head of Department of Information Technology for guiding and encouraging us throughout the project development.

With pleasure, I would also like to thank all the friends who have contributed directly or indirectly to my internship.

Abstract

The internship was completed at the Kathmandu-based SimplySoft Technologies Pvt Ltd in Dillibazar. Testing and analyzing new functionality for the production environment as well as finishing the allocated project are some of the ways I contribute to the organization. AngularJS was used to build the whole application known as Revenue Platform. The project was created with the purpose of replacing the existing platform, and it also includes a basic backend built using MongoDB and communicating over REST API. According to the organization's criteria, this project was developed utilizing leading technologies.

Keywords: revenue platform, AngularJS, ChartJS, API, NoSQL

Table of content

Acknowledgement	i
Abstract	ii
Table of content	iii
List of Abbreviations	v
List of Figures	vi
List of Tables	viii
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Scope and Limitations	2
1.5 Report Organization	2
CHAPTER 2: ORGANIZATION DETAILS	4
2.1 Introduction to Organization	4
2.2 Organization Hierarchy	5
2.3 Working Domain of Organization	6
2.4 Description of Intern Department	7
2.5 Literature Review	8
CHAPTER 3: INTERNSHIP ACTIVITIES	10
3.1 Roles and Responsibilities	10
3.2 Weekly Log	10

3.3 Description of the Project	12
3.3.1 Requirement Analysis	13
3.3.2 Feasibility Study	15
3.4 Activities Performed	16
3.4.1 Designing UI	18
3.4.2 Creating the Home Page	19
3.4.3 Creating the Database	20
3.4.4 Creating the API	21
3.4.5 Development of Creating Receipt Page	22
3.4.6 Developing the View for Paid and Unpaid Vouchers	24
3.4.7 Developing Search by PAN page	28
3.4.8 Developing Analysis Page	28
CHAPTER 4: CONCLUSION AND LEARNING OUTCOMES	34
4.1 Conclusion	34
4.2 Learning Outcome	34
References	35
Appendices	36

List of Abbreviations

API: Application Programming Interface

CRUD: Create Read Update Delete

FCGO: Financial Comptroller General Office

PAN: Permanent Account Number

REST API: Representational State Transfer API

UI: User Interface

UX: User experience

List of Figures

Figure 2.1: Organization hierarchy of SimplySoft Technologies	6
Figure 3.1: System Functions of Revenue Platform	13
Figure 3.2: Use case diagram of the system	14
Figure 3.3: Home page of the eCommerce Platform	17
Figure 3.4: Add and remove to cart functionality	17
Figure 3.5: Iterative loop followed during the internship	18
Figure 3.6: Wireframe of the homepage	18
Figure 3.7: Wireframe of the form page	19
Figure 3.8: Homepage of the Application	19
Figure 3.9: District data stored in the database	20
Figure 3.10: Sample PAN data stored in the Database	21
Figure 3.11: Different Requests of the API	21
Figure 3.12: Selecting Province	22
Figure 3.13: Selecting District name	22
Figure 3.14: Selection of District and Office Name	23
Figure 3.15: Receipt Creation page	24
Figure 3.16: Receipt Data Entry Successful	24
Figure 3.17: Paid voucher view	25
Figure 3.18: Paid Voucher Preview page	25
Figure 3.19: Voucher Print Page	26
Figure 3.20: Unpaid View page	26

Figure 3.21: Unpaid voucher preview page	27
Figure 3.22: Voucher payment page	27
Figure 3.23: Search by PAN page	28
Figure 3.24: Response from API call	29
Figure 3.25: Analysis Page	31

List of Tables

Table 2.1: Company Details	5
Table 2.2: Internship Details	8
Table 3.1: Weekly Log	11
Table 3.2: Unit Testing on Reading Payment Status Data from API	32
Table 3.3: Unit Testing on Search by PAN Function	32
Table 3.4: Unit Testing on Create Receipt Function	33

CHAPTER 1: INTRODUCTION

1.1 Introduction

Revenue Platform is a centralized online tax payment and management portal. It is a web-based application that has a centralized UI and modern database support. Revenue Platforms can be used by people, companies, and organizations to pay for taxes and services. The revenue platform also provides details to the federal government about tax payment information and details regarding the revenue generated. It also provides other functionalities like dynamic analysis of data, generation of individual payment receipts. This platform is built to enhance and upgrade the functionality of the current platform.

The internship period at SimplySoft Technology Pvt. Ltd last from 27 Chaitra 2078 to 24 Asadha 2079 as an Angular developer and was involved in developing an internal iteration of the **Revenue Platform** under the mentorship of Mr. Kunjan Pandey, the Managing Director at SimplySoft. Revenue Platform is one of the products of SimplySoft that is being built for the Financial Controller General Office (FCGO). The project was to build an internal clone of the production environment from scratch using different UI components and databases in order to internally test enhancements, new functionality, and new technologies.

SimplySoft is a full-cycle software engineering and development company that provides high-quality software and development services by combining industry best practices in business process analysis and modeling with world-class technological expertise. The organization is exclusively focused on the client, with a strong dedication to fully understanding and resolving customer difficulties. SimplySoft's goal is to help our customers become more agile and lucrative. The company's objective is to supply customers with exceptional IT expertise and cutting-edge software solutions. The organization closely works with government agencies in order to facilitate and develop technologies used to implement e-governance.

1.2 Problem Statement

Currently, there is no effective way in which revenue is managed properly.

The main problems faced by current revenue management systems are as follows:

- Manual Entry of payments and invoice generated in paper form.
- Difficult to maintain accurate details such as PAN number, tax details, etc.
- No centralized database for indexing and managing all the different revenue portals

1.3 Objectives

The objective of the internship project is:

- To design and develop a revenue platform system for automatic voucher creating and analysis using AngularJS.

1.4 Scope and Limitations

Scope:

The scope of this project is to be a center for all things related to revenue management. SimplySoft aims to use this project as an experimental platform, where they can test different functions and new technologies, and use it as a debugging tool where they can test different security vulnerabilities. They aim to use this project as a training tool, to train the intern to be familiar with the tech stack and create a learning environment.

Limitation:

The current limitation of this project is the lack of security mechanisms, especially in the backend API. API design and development were not given much focus due to which only a basic REST API is formed. Some other features that are to be improvised are more detailed analysis and trend prediction in the analysis section, a more dynamic architecture for the system, and addition of more UI components.

1.5 Report Organization

Chapter 1: Introduction

This section describes the overall project concept in brief. It contains all the details like an introduction to the project, its problem statements and objectives, scope and limitation of the project along with the report organization section.

Chapter 2: Organization Details

This section contains the details of the Organization. It is composed of in-depth mention of the organization's structure, its operating domain, and the hierarchy of the organization. It also describes the intern's department.

Chapter 3: Internship Activities

This Section includes the internship activities with details of my roles and responsibilities, a weekly log, and a description of my tasks performed, and the project done in-depth.

Chapter 4: Conclusion and Learning Outcomes

This section includes the things learned during the internship tenure. It also contains the evaluation of the report, findings, learnings, and future recommendations.

Appendix and References

Appendix describes the canvases of the system and includes screenshots and some code snippets developed during the internship.

References contain information regarding all the work in this project other than the authors' and they have been carefully cited and referenced. Sources include books, websites, journals, magazines, case studies, and thesis documents.

CHAPTER 2: ORGANIZATION DETAILS

2.1 Introduction to Organization

SimplySoft is a full-cycle software engineering and development company that provides high-quality software and development services by combining industry best practices in business process analysis and modeling with world-class technological expertise. It is located at Dillibazar, Kathmandu. It was established in 2005. The organization is exclusively focused on the client, with a strong dedication to fully understanding and resolving customer difficulties. SimplySoft's goal is to help our customers become more agile and lucrative. The company's objective is to supply customers with exceptional IT expertise and cutting-edge software solutions.

SimplySoft is one of the leading Solution providers in Nepal. It offers IT outsourcing services covering the whole spectrum of clients' needs: full-cycle software development; business analysis and software design; software systems integration and middleware; software systems migration; software conversions and enhancements; software re-engineering; software testing; software implementation and support.

While working in Nepal and foreign IT markets, the company is 100 percent customer-driven and fully committed to thoroughly understanding and solving customers' problems. SimplySoft's mission is to make its customers more agile and profitable. The goal of the company's activity is to provide customers with advanced IT expertise and cutting-edge software solutions. (SimplySoft,2021)

SimplySoft believes in the probability of Information and Communication Technology (ICT) as a secure and rewarding experience as well as a powerful tool for both economic and infrastructure development. The services provided by SimplySoft are

- IT Outsourcing Services
- Software Development & Solutions
- Software System Integration
- Software System Migration
- Software Testing/Implementation/Support

- Backup, Archiving & Disaster Recovery Services
- Database Security
- Corporate Training
- Network Security

Exceeding expectations, understanding its clients' businesses, and designing robust high-quality solutions at surprisingly affordable costs are just a few of the key differentiators that set SimplySoft apart from other software development firms.

Table 2.1: Company Details

Company Name	SimplySoft Technology Pvt. Ltd
Location	Dillibazar, Kathmandu, Nepal
Established Date	2005
Number of Employees	15
Website	https://simplysoft.technology/

2.2 Organization Hierarchy

SimplySoft has a very small but robust team of 15 members. Each of the members is very specialized in their respective domain and everyone works collaboratively to achieve the common goal. The CEO/founder leads the team as a Project Manager. He delegates tasks, supervises and oversees any business development activities, and is also the Database Administrator for all the projects.

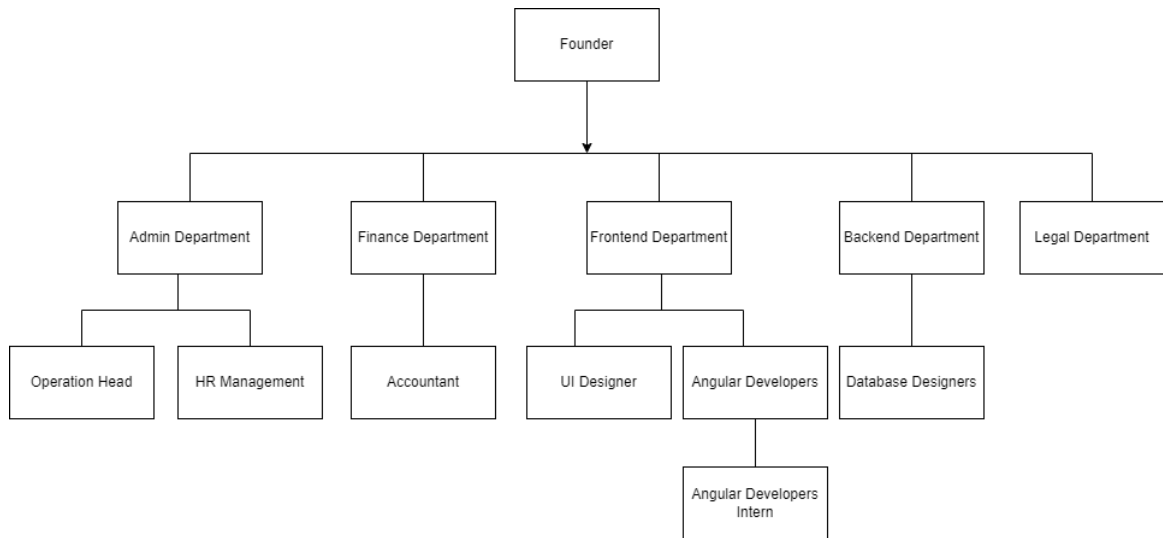


Fig 2.1: Organization hierarchy of SimplySoft Technologies

The angular intern falls under the frontend department and is supervised by mid and senior-level developers in the department. The intern is assigned tasks and projects by the team. The intern interacts daily with all the members of the frontend team and is continuously learning and implementing things with guidance from the team. The intern reports directly to the project manager.

2.3 Working Domain of Organization

SimplySoft is a fully featured software development company that focuses mainly on Governmental projects, and large-scale software development projects. They work on Java, Oracle, Angular, and Asp.Net. It has a wide range of commercial clients and has completed many governmental projects which are used daily by thousands of users.

Some areas that the organization works on (SimplySoft, 2021) are

- full-cycle software development,
- business analysis, and software design
- software systems integration and middleware
- software systems migration
- software re-engineering

- software testing
- software implementation, and support

While working in Nepal and foreign IT markets, the company has deployed a wide spectrum of information systems for the finance, production, transportation industries, as well as healthcare and education services. Some popular clients (SimplySoft, 2022) whose projects that SimplySoft has developed are

- CGAS: Ministry of Finance (FCGO)
- RMIS: Office of the Comptroller and Auditor General
- Council for Technical Education and Vocational Training
- Nepal Bank Limited
- Rural Water Supply & Sanitation Fund Development Board
- Butwal Power Co. Ltd.

2.4 Description of Intern Department

SimplySoft has different departments each having its own function and objective. SimplySoft generally only tackles one major project at a time and each department is assigned its own objectives and deliverables. All of these departments take part in the whole process which starts long before the actual code is written. Regular syncs with the client take place where the client and all the different teams come together to talk about progress and if there are any blockers. The client requirements can also change based on different factors which are then reflected in the development cycle.

As shown in figure 1, the Angular intern is a part of angular developers under the frontend department. This department is responsible for designing and developing a user-friendly UX/UI which also meets the client's requirements and technological feasibility. The designer and developers work together to plan and lay out the road map to create the front end. The UI designer develops a UI for the project, and it is revised and updated after taking feedback from the development team. Then, the front-end development team starts developing the UI. The intern is given small tasks like researching functionalities, and testing code in their local environment. The intern is also sometimes tasked with developing

small functionalities which then later get pushed to production after being reviewed by the senior dev. The intern is also provided with a project which they need to design and develop from scratch. By tallying with the features provided by the client the design is to be finalized and after that, the design is brought to life in the front-end department.

Table 2.2: Internship Details

Company Name	SimplySoft Technology Pvt. Ltd
Internship Position	Angular Intern
Start Date	27 Chaitra, 2078
End Date	24 Ashar, 2079
Mentor	Mr. Kunjan Pandey
Working Days	Sunday to Friday
Working Hours	11 AM -5:30 PM

2.5 Literature Review

AngularJS

Angular JS was used to create the frontend UI of the application. AngularJS was developed by Google Inc. with its first release in 2010. The motivation for the creation of AngularJS is to extend the HTML vocabulary with the use of data bindings through directives. At a high level, directives can be thought of as markers on a DOM (Document Object Model) element such as attributes, element name, comments, or CSS class that tells the HTML compiler (\$compile) within AngularJS to attach a specific behavior to that DOM element (via event listeners) or transform specific DOM elements and propagate these changes to

its children. This is implemented with the use of ng-tags, which bind the view to one or many models. In addition, AngularJS includes test suites which make it easier to test individual components with built-in dependency injection. Finally, AngularJS is implemented using the MVC (Model-View-Controller) architecture concept where parts of an app are isolated such that the application logic is isolated from the user interface. An example of applications built with AngularJS is Netflix and YouTube for PlayStation. (Sufyan bin Uzayr , 2019)

CHAPTER 3: INTERNSHIP ACTIVITIES

3.1 Roles and Responsibilities

During my role as an Angular intern, I was assigned various duties throughout the course of three months. I was responsible for creating the internal version of the application and adhering to the standards set by the organization. I was tasked with creating the whole application using the Angular framework.

My supervisor Mr. Kunjan Pandey, the Managing Director at SimplySoft assigned me tasks and also evaluated me on a daily basis. I also worked closely with the Senior Angular Developer Suraj Sir and Roshan Sir during the development of my project. My roles and responsibilities during my internship period were as follows:

- Learn about the Angular Framework
- Learned about routing and integrating services in an Angular application
- Aid in the research and development of modules in the production environment.
- Create the internal version of the application
- Learn about and create a database for the internal application using MongoDB
- Create a REST API to fetch data from the Database
- Use the API to automatically populate the frontend
- POST data from the frontend to the database

3.2 Weekly Log

Below is the weekly log of my internship period at SimplySoft Technology Pvt. Ltd:

Table 3.1: Weekly Log

Week	Task	Outcome
1	Knowing the organization workflow and reviewing basics of HTML, CSS. Getting started with AngularJS.	Got the basic concept of AngularJs and its framework.
2	Got to know the working of components, lifecycle hooks and binding.	Made a hello world application incorporating the knowledge.
3	Learned about dependency injection, services, routing and navigation.	Used pipes, HTTP and Observables.
4	Learned about Form creation and validation along with creating add-to-cart applications.	Used all learned domains in making a reactive add-to-card application.
5	Coded to generate the old UI of the revenue platform along with designing a new layout for the platform.	Determining technical issues, researching new UI designs feasible for the project.
6	Generated different components required for the platform along with creating databases.	Made JSON files and fed the database.
7	Made a service page (main.service.ts) and connected with the database API created.	Linked components to database using API.

8	Added new components along with debugging the errors generated.	Worked on the feedback provided by the mentor after a demo.
9	Generated form page and POST data in database along with debugging error.	Database schema restructured along with adding new features in the UI.
10	Analysis performed from the database data along with testing and debugging.	Demo presentation along with generating new API for analysis.

3.3 Description of the Project

Revenue Platform is a centralized online tax payment and management portal. It is a government project given by the FCGO to SimplySoft which is currently in the production phase. It is a web-based application that is used internally in the ministry.

Revenue Platforms can be used by people, companies, and organizations to pay for taxes and services. The revenue platform also provides details to the federal government about tax payment information and details regarding the revenue generated. It has multiple functionalities like

- Create a new entry for payment
- Search by Pan Number
- Distinct entry form for every district office
- Automatic retrieval of paid and unpaid details
- Report log based on analysis done on database
- Generation of payment receipt digitally dynamically
- New UI components based on new standards

- NoSQL database with full REST API support

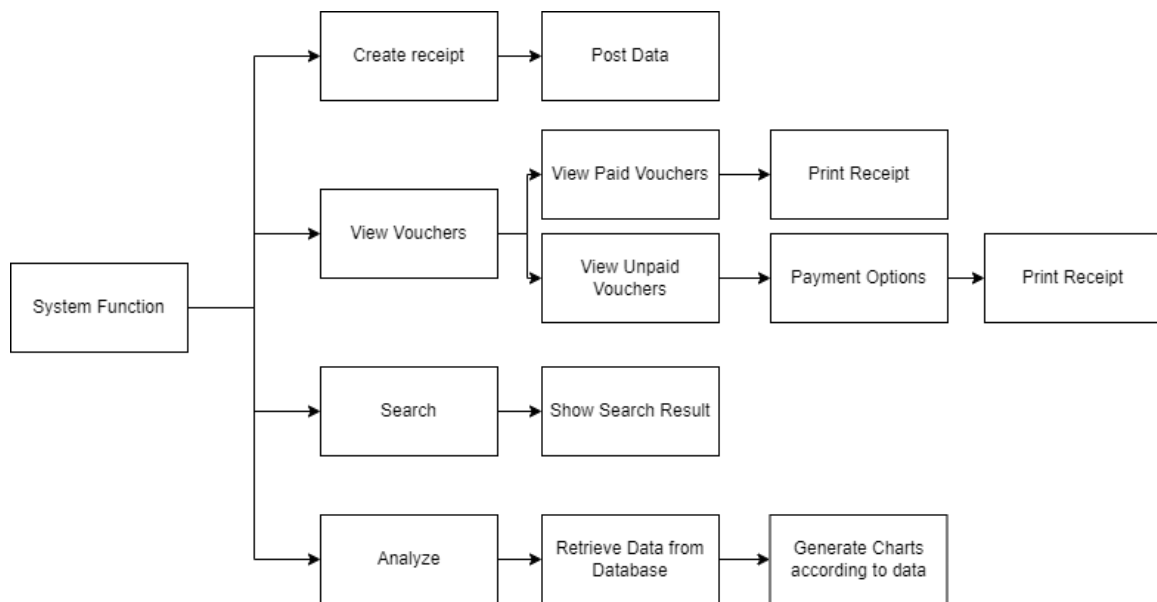


Figure 3.1: System Functions of Revenue Platform

The above figure describes the architecture and flow of the application. The application is made entirely using the Angular framework. It is able to post data into the database via its receipt creation form. The application is also able to extract data from the database by using the different routes that are defined in the API interface. It also supports the dynamic generation of receipts after payment is done by a user. Since this application is developed for internal testing and experimental purposes, it is fed dummy data and the payment is not redirected to the payment gateways.

3.3.1 Requirement Analysis

Following are the requirements for this project.

Functional requirement

System functional requirements describe activities and services that the application provides.

- Users can search and view details using PAN numbers.
- Users can create a new receipt for the voucher.

- Users can view the vouchers according to their payment status
- Users are available to access the analysis regarding the different data stored in the database

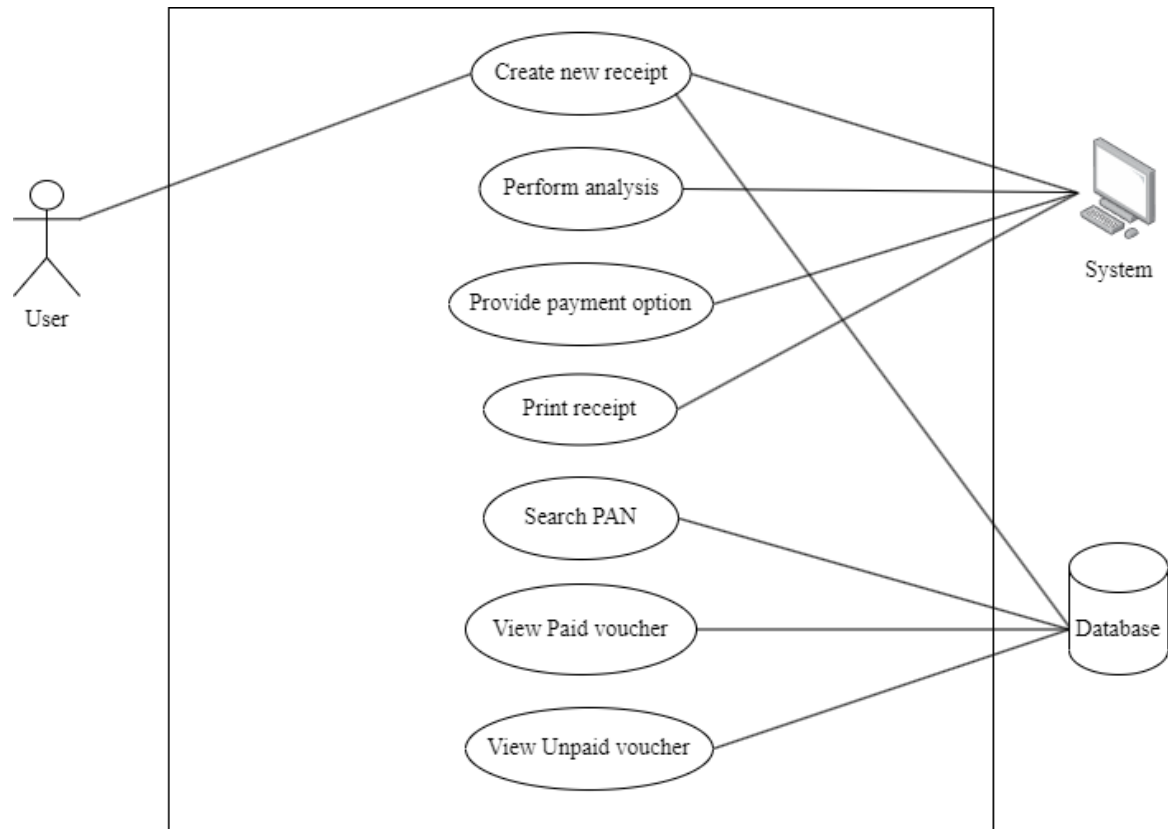


Figure 3.2: Use case diagram of the system

A user can create a new receipt using the form provided. They can also view an analytical report of the data that is stored in the database. They can also search for a user via their PAN number. They can view the details of paid and unpaid vouchers of different people. The user is also able to pay for the services and print dynamic receipts for each transaction.

Non-Functional Requirement

Non-functional Requirements are characteristics or attributes of the system that can judge its operation. The following points clarify them:

- Accuracy and Precision: The system was able to perform its process with accuracy and precision and was able to avoid problems.
- Flexibility: The system was easy to modify, and errors were easily fixed.

- Security: The system is secured and has preserved the user's information.
- Usability: The system is made so that it is easily understandable and dealt with.
- Maintainability: The maintenance process can cope with any problem that arises suddenly.
- Speed and Responsiveness: The system was quick to respond to the user's requests.
- Scalability: The system can scale according to the needs.

3.3.2 Feasibility Study

Feasibility studies aim to uncover the strengths and weaknesses of an existing or proposed system, opportunities and threats objectively and rationally as presented by the environment, the resources required to carry through, and ultimately the prospects for success. The feasibility study of this application had been carried out which is as follows:

Technical Feasibility

There are not any technical constraints that hampered the project development and resulting software. All the technologies and databases used in the making can be accessed for free.

Operational Feasibility

The project was implemented in a way that allowed the functioning of revenue payment smoothly. It also provides a user-friendly interface in a modular fashion. The operation of the project runs smoothly in a controlled environment.

Economic Feasibility

Development of this application is safest from economic hardships as this project is free from financial contribution. The technologies used can be easily accessed for free to launch the first prototype. But this advantage is a short-term one as further modification of the application (addition of new servers, data storage, scaling of the application) would increase the price exponentially.

Schedule Feasibility

The project was completed in the scheduled time. Only small changes to strategic plans had been made for the completion of this project.

3.4 Activities Performed

As an Angular Developer Intern, various tasks were assigned ranging from building simple functions such as an automatic date converter to testing and researching new methodologies to perform any specific function.

The tools used during the internship and the duration of the project were:

1. AngularJS: It was used to create the frontend based on the Figma design and to interact with the backend
2. NodeJS: It was used to design and develop the REST API to interact with the database
3. MongoDB: It was used to store and retrieve data using the REST API.
4. Git: It was used for version control.
5. Visual Studio Code: It was the primary IDE used for the development of the project.
6. Figma: It was used to create the wireframe for the front end.
7. Draw.io: It was used to create the different diagrams for this report.

During the beginning of the internship, a fair amount of time was spent researching and learning about the Angular Framework and the different technologies that it relies on. Small and simple tasks were performed to be familiar with the Angular Framework. Beginner projects were built by following some tutorials as well as by researching and implementing certain concepts.

One of the first tasks performed was to build a simple eCommerce platform that supported adding to and deleting from cart functionality. This was built to better understand how components worked, how routing worked between different components, and to better understand the CRUD operation on the Angular Framework.

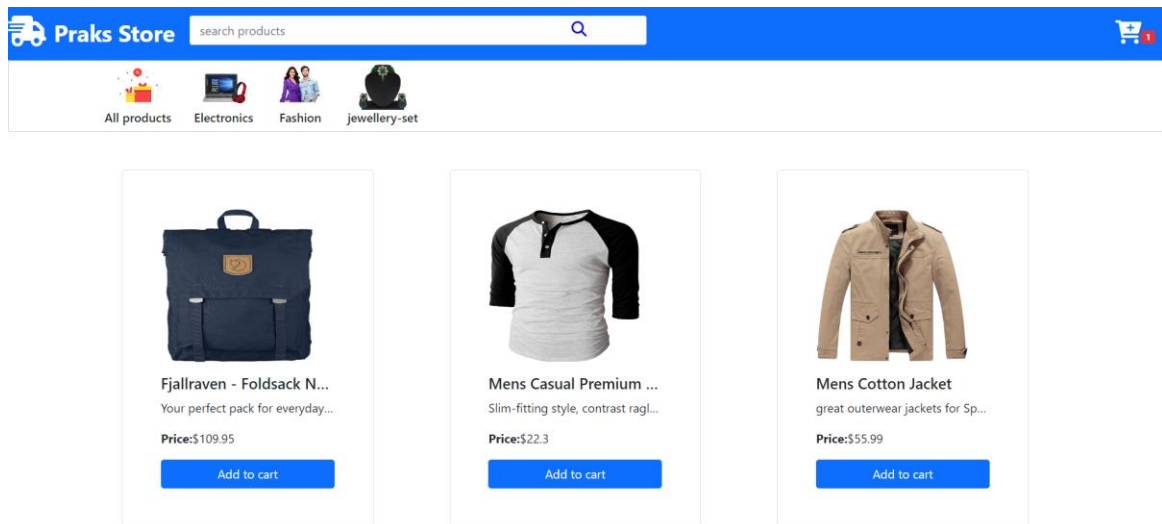


Figure 3.3: Home page of the eCommerce Platform

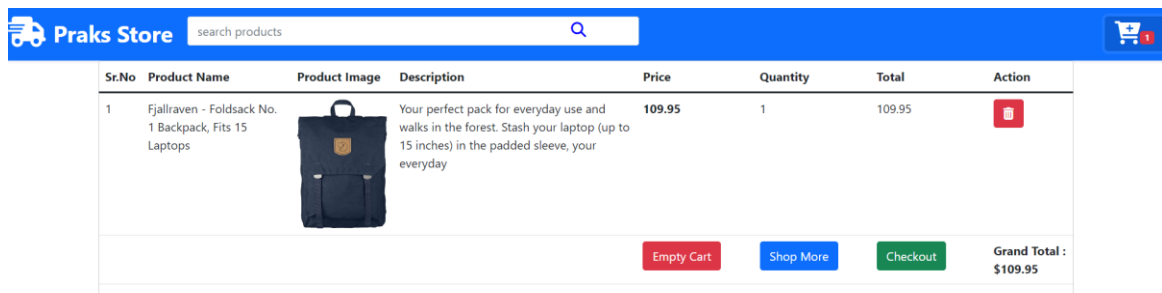


Figure 3.4: Add and remove to cart functionality.

The main project that was assigned was to build a Revenue platform for an internal copy of the production environment from scratch. Firstly, an exact copy of the environment was built, and feedback was received to make it more scalable and to implement code reusability. A second iteration of the front end was built considering the feedback from the supervisor. The supervisor reviewed and provided guidance on how it can be improved. The organization followed the agile framework so there was a continuous feedback loop.

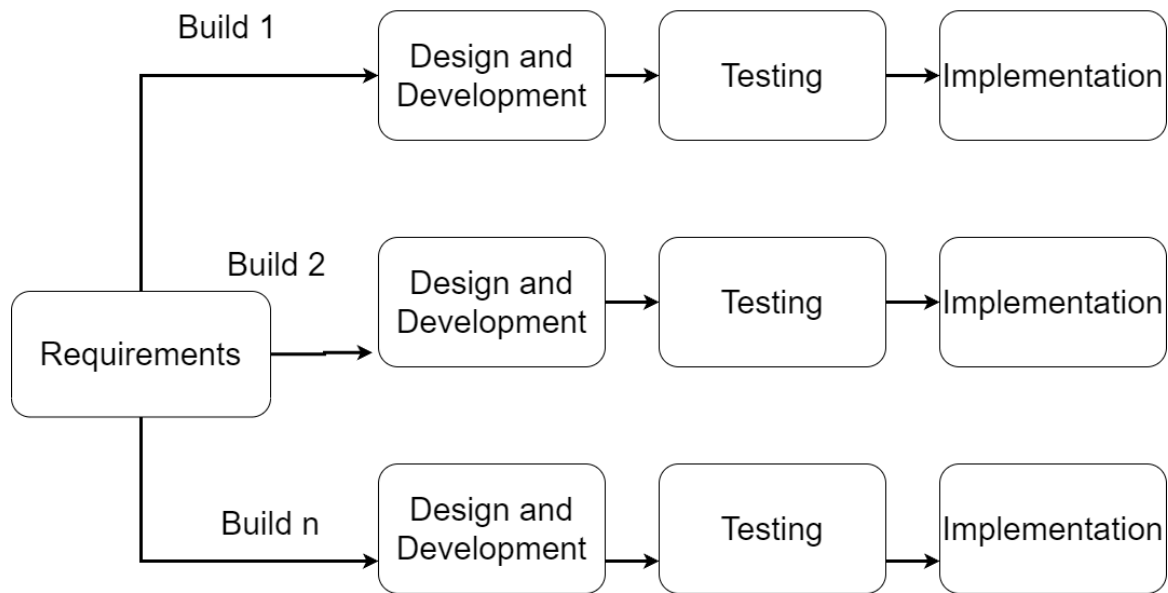


Figure 3.5: Iterative loop followed during the internship.

Finally, the task to create the full application using modern frameworks like Tailwind CSS, MongoDB was assigned.

The activities performed during building this project are as follows:

3.4.1 Designing UI

The UI for the application was built using Figma. The guidelines for the UI were provided by the organization. A wireframe mockup was built for the Homepage and the form entry page.



Figure 3.6: Wireframe of the homepage

Government of Nepal
Ministry of Finance
Financial Controller General Office (FCGO)

flag

Receipt Entry

Receipt Details

English Date*

Bank* Revenue Office*

Payers Details

Payers Code PAN English Name

Address Phone Number

S.No.	Purpose*	Revenue Head*	Description	Amount	Tax Year	Payment Status	Remarks

Cancel Next

Figure 3.7: Wireframe of the form page.

3.4.2 Creating the Home Page

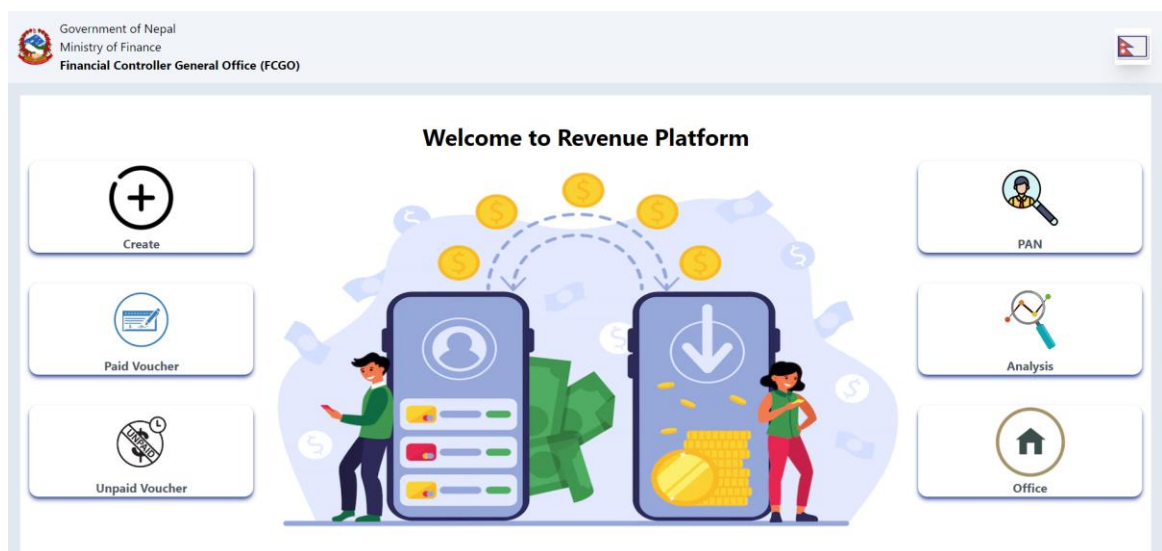


Figure 3.8: Homepage of the Application

The home of the application was created following the wireframe that was designed. It served as the main hub for the application from where the user can navigate to the different parts of the application. Since AngularJS follows MVC architecture, a separate component called homepage was created where different files attaining to the MVC architecture are created automatically.

The command `ng generate component <component name>` is used to automatically generate files required for the component.

3.4.3 Creating the Database

A database was created on MongoDB in which data regarding the different offices and their location are stored. The data is stored in a Nested JSON Format. The data is retrieved from the database using a route specified in the API. Another collection was created to store the data related to user information, such as pan number, address, payment status, amount, name, phone number, revenue head, purpose, etc. All these fields will be populated by the data which is sent from the application's create receipt page.

The database contains two documents: pandata and location. These are then later queried by the application.

```
_id: ObjectId('62d557cbc7549b6413e84e3b')
pname: "province6"
districtDetails: Array
  0: Object
    DistrictName: "Rukum"
    name: Array
      0: "204975901-Rukum Jilla Adalat"
      1: "312005901-Pradhanmantri krishi adhunikikaran pariyojana priyojana kary..."
  1: Object
    DistrictName: "Salyan"
    name: Array
      0: "212916001-District Election Office Salyan"
      1: "301076001-Rastriya Anusandhan Jilla Karyalaya Salyan"
  2: Object
    DistrictName: "Dolpa"
    name: Array
      0: "305016101-ko le ni ka"
      1: "314926102-District Administration Office"
  3: Object
    DistrictName: "Jumla"
    name: Array
      0: "204916201-High Court Surkhet, Jumla Bench"
      1: "210016201-Public service commission, Jumla office"
  4: Object
    DistrictName: "Mugu"
    name: Array
      0: "329063301-Rara national park"
      1: "314926302-DAO MUGU"
  5: Object
  6: Object
  7: Object
  8: Object
  9: Object
```

Figure 3.9: District data stored in the database.

```

_id: ObjectId('6314d569ac5fc1563a7811ad')
fiscalYear: 2078
payersCode: 4568
agencyName: "305043001-Taxpayer Service Office, galchhi"
date: 2022-06-09T00:00:00.000+00:00
panNumber: 312345678
bank: "Nepal Bank Limited"
engName: "Sujan Shah"
npName: ""
address: "Jitpursimara, Bara"
phoneNumber: 9841000040
purpose: "Service Pay"
revenueHead: "Badhfadh prapta hune ghar jagga registration dastoor"
description: "testing description 4"
amount: 4300
remarks: " Testing Data 4"
paymentStatus: "paid"

```

Figure 3.10: Sample PAN data stored in the Database.

3.4.4 Creating the API

A simple REST API was developed to interact between the application and the database. This API was developed by using NodeJS. Different routes and GET and POST requests were defined, each having its own response format and functionalities. The Angular frontend utilized service calls to query, save and display data to and from the database. The database uses HTTP and HTTPS protocols to perform the request-response cycle.

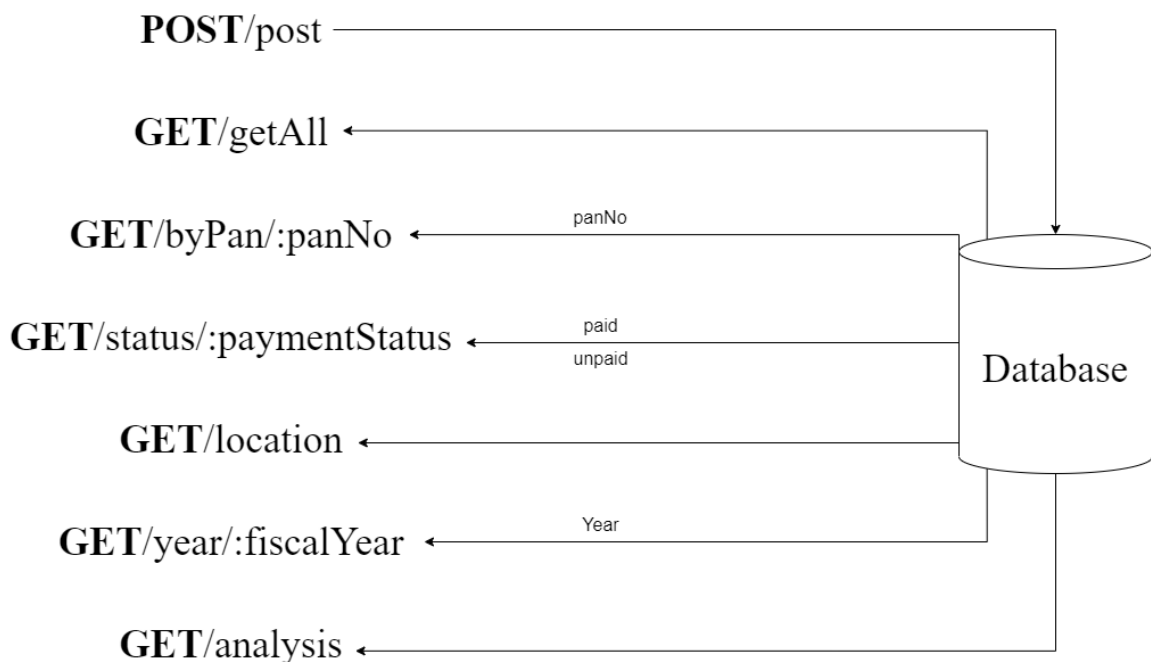


Figure 3.11: Different Requests of the API

3.4.5 Development of Creating Receipt Page

This page provides the functionality of creating new payment receipts for any customer. The user needs to select the province, district, and office from which payment is being done. The user can also search for the district if they chose to. Then the user is redirected to the receipt form page. It consists of different fields where the personal details of the customer are entered. The payment status and amount can also be specified. The design of the form follows the wireframe which was made following the guidelines of the project. After filling out the form, the data is stored in the database. The page provides the option of selecting the Federal Government as a whole consisting of all districts or selecting the district on the basis of the province.

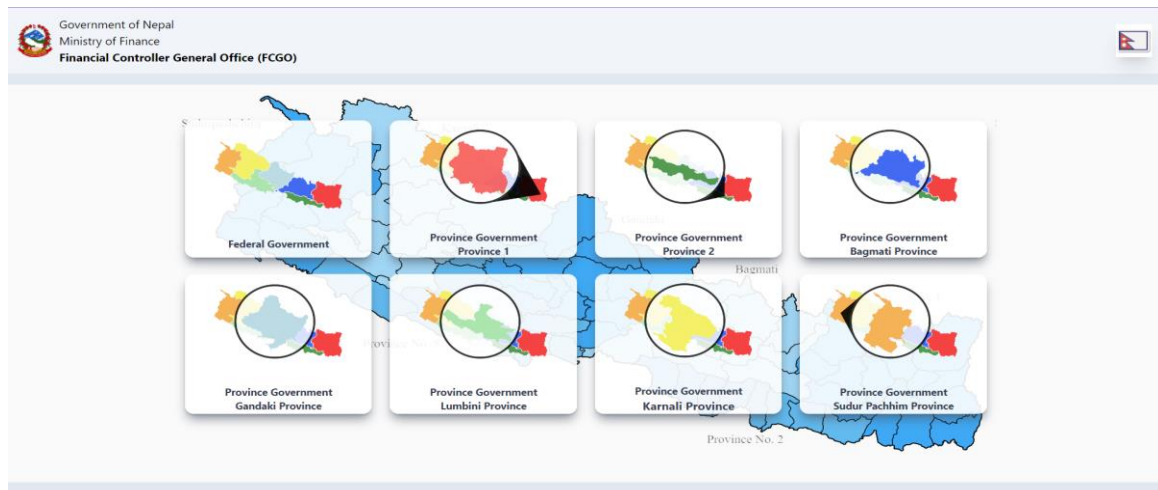


Figure 3.12: Selecting Province

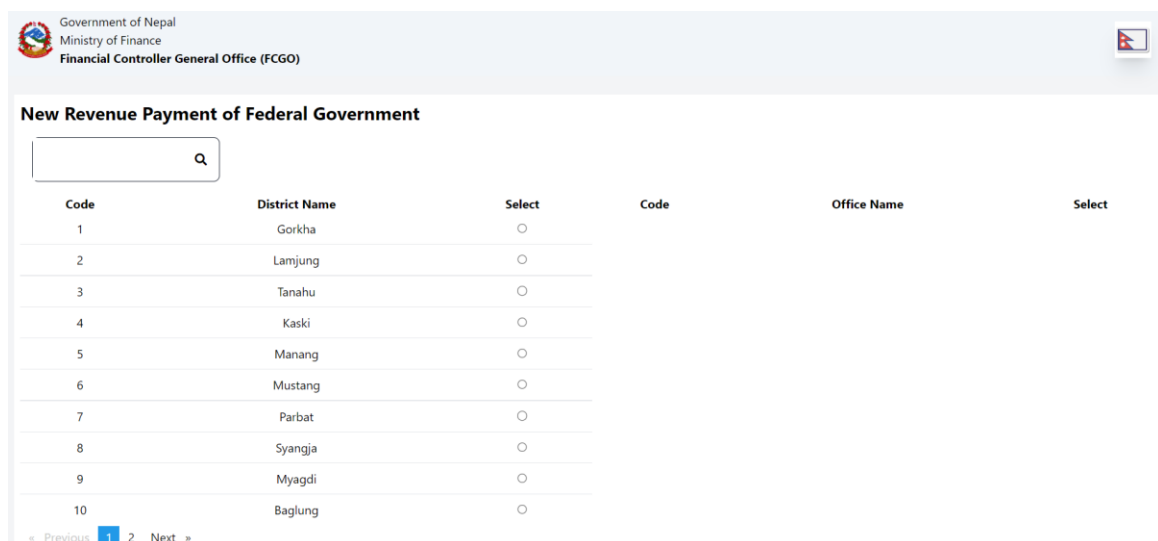


Figure 3.13: Selecting District name

Only after selecting the district name, does the office data populate in regard to the selected district. This is achieved due to the validation that was implemented while creating the page. The data is dynamically fetched from the database.

After selecting the district and the office name, the Next button option is visible due to validation used in the function. After selecting the next button, the user is directed to the form page where the user needs to provide the information as needed such as PAN number, name, address, revenue head etc. The form page UI is created from the wireframe design. After entering all the required fields, the next button present in the form page POST the data to the database and shows a Data Added Successfully message on the page.

The screenshot shows the 'New Revenue Payment of Federal Government' form. At the top, there is a search bar with a magnifying glass icon. Below it is a table with columns: Code, District Name, Select, Code, Office Name, and Select. The 'Select' column contains radio buttons, with the one for 'Manang' selected. The 'Office Name' column shows two options: '204974001-Manang District Court' and '312004001-Project Implementation Unit- Apple Zone'. A 'Next' button is visible on the right side of the table.

Code	District Name	Select	Code	Office Name	Select
1	Gorkha	<input type="radio"/>			<input type="radio"/>
2	Lamjung	<input type="radio"/>			<input type="radio"/>
3	Tanahu	<input type="radio"/>	1	204974001-Manang District Court	<input checked="" type="radio"/>
4	Kaski	<input type="radio"/>			<input type="radio"/>
5	Manang	<input checked="" type="radio"/>			<input type="radio"/>
6	Mustang	<input type="radio"/>			<input type="radio"/>
7	Parbat	<input type="radio"/>			<input type="radio"/>
8	Syangja	<input type="radio"/>			<input type="radio"/>
9	Myagdi	<input type="radio"/>	2	312004001-Project Implementation Unit- Apple Zone	<input type="radio"/>

Figure 3.14: Selection of District and Office Name

The screenshot shows the 'Receipt Entry' form. It is divided into two main sections: 'Receipt Details' and 'Payers Details'.
Receipt Details:
 - ENGLISH DATE *: mm/dd/yyyy
 - BANK*: [Empty field]
 - REVENUE OFFICE *: 204974001-Manang District Court
Payers Details:
 - PAYERS CODE: [Empty field]
 - PAN: [Empty field]
 - ENGLISH NAME *: Prakriti Bista
 - ADDRESS *: Basundhara, Kathmandu
 - PHONE NUMBER*: 9860485617

S.No.	Purpose*	Revenue Head*	Description	Amount*	Tax Year	Payment Status	Remarks
1.				XXXX			

Figure 3.15: Receipt Creation page

Government of Nepal
 Ministry of Finance
 Financial Controller General Office (FCGO)

Data Added Successfully

Receipt Entry

Receipt Details

ENGLISH DATE *

BANK*

REVENUE OFFICE *

Payers Details

PAYERS CODE PAN ENGLISH NAME *

Figure 3.16: Receipt Data Entry Successful

3.4.6 Developing the View for Paid and Unpaid Vouchers

Individual views for viewing the paid and unpaid vouchers were developed. The data for the views are dynamically fetched from the API.

API call for Paid data: <http://localhost:3000/api/status/paid>

API call for Unpaid data: <http://localhost:3000/api/status/unpaid>

The data from the API calls are stored and then dynamically displayed on the layout.

Government of Nepal Ministry of Finance Financial Controller General Office (FCGO)								
Paid Voucher								
TRANSACTION CODE	FISCAL YEAR	AGENCY NAME	DATE	PAN NUMBER	NAME	AMOUNT	STATUS	ACTION
1	2078	305043001-Taxpayer Service Office, galchhi	2022-06-09T00:00:00.000Z	312345678	Sujan Shah	4300	PAID	Preview
2	2078	01014901-District Lawer Office Kapilvastu	2022-06-09T00:00:00.000Z	512345678	Veerata Rana	3300	PAID	Preview
3	2078	308033103-Chitwan Underground Coordinative Water Irrigation Project	2022-06-09T00:00:00.000Z	112345678	Anish Tamang	2000	PAID	Preview
4	2078	301016601-District Government Attorney Office Jajarkot	2022-06-09T00:00:00.000Z	712345678	Prabin Karki	12000	PAID	Preview
5	2078	301077601-National Investigation District Office Kanchanpur	2022-09-05T05:41:48.526Z	123456111	Aakash Sarraf	2100	PAID	Preview
6	2075	336015501-Malpot Karyalaya Khalanga Pyuthan	2022-09-05T07:46:16.276Z	213243545	Asmi Sharma	23000	PAID	Preview
7	2076	01014901-District Lawer Office Kapilvastu	2022-09-05T07:54:03.476Z	221222333	Jina Gurung	9800	PAID	Preview

Figure 3.17: Paid voucher view

The user can select a preview to individually view the voucher. They can also print the receipt for every paid voucher.

Government of Nepal Ministry of Finance Financial Controller General Office (FCGO)				
Government Revenue System 305043001-Taxpayer Service Office, galchhi	Code 1 English Date 2022-06-09T00:00:00.000Z Total Amount "Rs ." 4300	Press Button for Print		
Payer Detail Sujan Shah 9841000040 Jitpursimara, Bara 312345678 Nepal Bank Limited				
S.No.	Revenue Head	Remarks	Amount	Tax Year
1	testing description 4	Testing Data 4	4300	2078

Figure 3.18: Paid Voucher Preview page

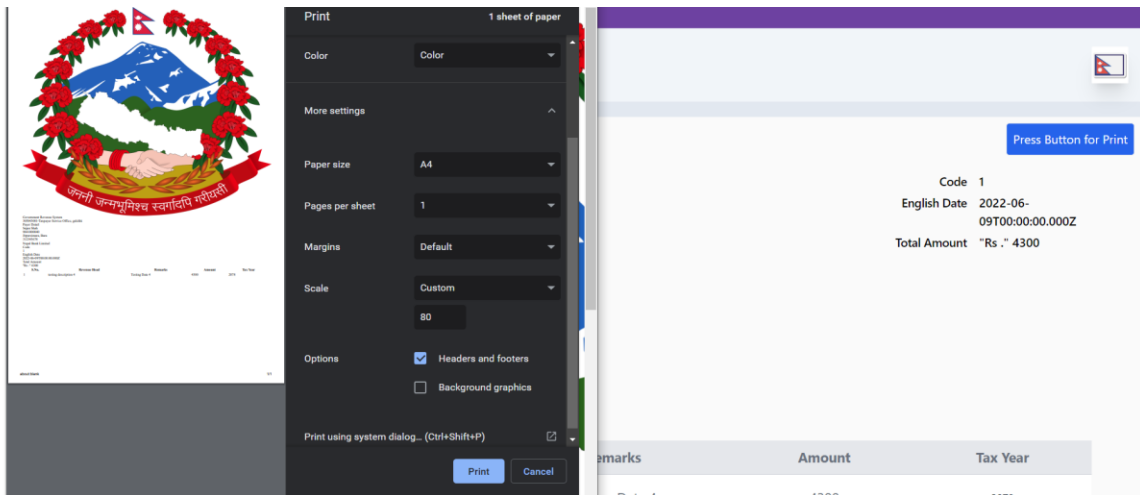


Figure 3.19: Voucher Print Page

Like the paid voucher page, the unpaid voucher also displays the data via a database where it displays only those data that have not been paid. As API is used to fetch the data, we can determine functionality to extract only the data needed for the display. The unpaid voucher page provides one additional feature compared to paid voucher, that is, voucher payment page which provides various getaways to pay the amount stated in the voucher.

Government of Nepal
Ministry of Finance
Financial Controller General Office (FCGO)

Unpaid Voucher

TRANSACTION CODE	FISCAL YEAR	AGENCY NAME	DATE	PAN NUMBER	NAME	AMOUNT	STATUS	ACTION
1	2078	305043301-Inland Revenue Office , Bhaktapur	2022-06-09T00:00:00.000Z	212345678	Anupama Maharjan	11000	UNPAID	Pay Now
2	2079	305010701-District Treasury Office	2022-06-09T00:00:00.000Z	123456789	Ram Sharma	1000	UNPAID	Pay Now
3	2078	314043704-DISTRICT POLICE OFFICE LAMJUNG	2022-06-09T00:00:00.000Z	412345678	Suman Yadav	1500	UNPAID	Pay Now
4	2078	336015501-Malpot Karyalaya Khalanga Pyurthan	2022-06-09T00:00:00.000Z	612345678	Bijaya Tripathi	5639	UNPAID	Pay Now
5	2079	301077601-National Investigation District Office Kanchanpur	2022-06-09T00:00:00.000Z	912345678	Sachin Pokhrel	19000	UNPAID	Pay Now
6	2079	301077601-National Investigation District Office Kanchanpur	2022-09-05T04:50:26.058Z	9123454569	Aarekaz Pokhrel	19000	UNPAID	Pay Now
7	2073	01014901-District Lawyer Office Kapilvastu	2022-09-05T05:45:04.119Z	987787812	Rasil Adhikari	21000	UNPAID	Pay Now

Figure 3.20: Unpaid View page

Here, the user can select the pay now option to pay the amount. A preview page containing the details of the selected payee is displayed from where the use can proceed to payment.

Government of Nepal
Ministry of Finance
Financial Controller General Office (FCGO)

Government Revenue System
305043301-Inland Revenue Office , Bhaktapur

Code 1
English Date 2022-06-09T00:00:00.000Z
Total Amount "Rs ." 11000

Proceed To Payment Press Button for Print

Payer Detail
Anupama Maharjan
9841000030
Suryabinayak, Bhaktapur
212345678
Global Bank Ltd

S.No.	Revenue Head	Remarks	Amount	Tax Year
1	testing description 3	Test	11000	2078

Figure 3.21: Unpaid voucher preview page

Government of Nepal
Ministry of Finance
Financial Controller General Office (FCGO)

नेपाल सरकार
अर्थ मन्त्रालय
आन्तरिक राजस्व विभाग

Voucher Details
Fiscal Year: 2078
Date: 2022-06-09T00:00:00.000Z
Office: 305043301-Inland Revenue Office , Bhaktapur
Bank: Global Bank Ltd

Payee Detail
PAN No: 212345678
Name: Anupama Maharjan
Address: Suryabinayak, Bhaktapur
Phone Number: 9841000030

VOUCHER NUMBER	TITLE DESCRIPTION	REVENUE HEAD	AMOUNT	INCOME YEAR	PURPOSE
1	testing description 3	यातायातक्षेत्रको आम्दानी	11000	2078	TAX Pay
Total			11000		

PAN No: 212345678
Name: Anupama Maharjan
Address: Suryabinayak, Bhaktapur
Phone Number: 9841000030

VOUCHER NUMBER	TITLE DESCRIPTION	REVENUE HEAD	AMOUNT	INCOME YEAR	PURPOSE
1	testing description 3	यातायातक्षेत्रको आम्दानी	11000	2078	TAX Pay
Total			11000		

Choose the Payment Option

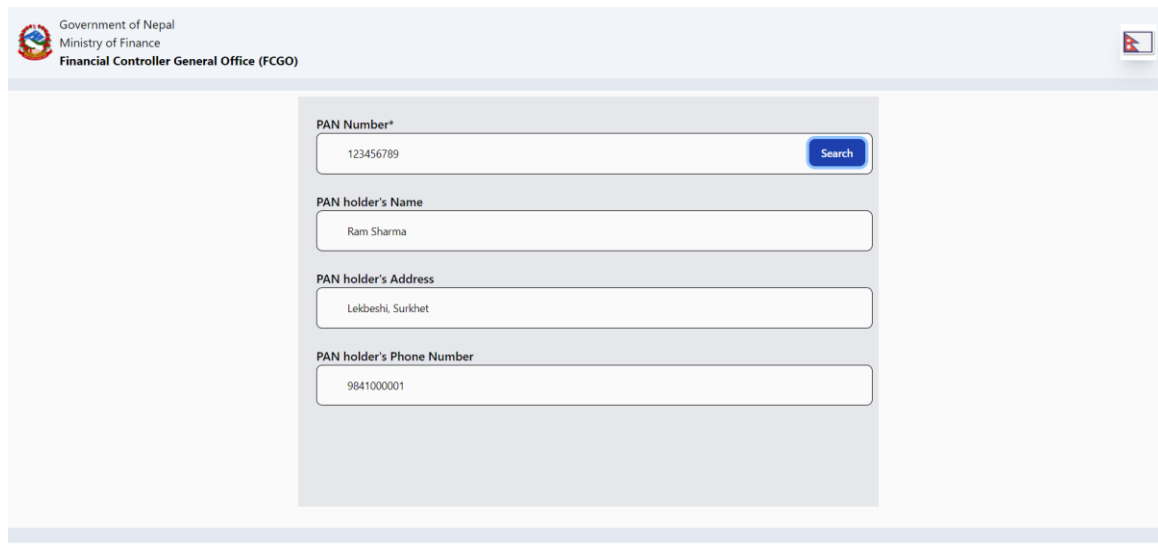
- connectIPS Popular
- eSewa
- FonePay
- Khalti
- Thaali

Figure 3.22: Voucher payment page

3.4.7 Developing Search by PAN page

This page provides the feature of searching for details via the pan number. A user can input a PAN number and search for details. The page then gets populated with the details of the user. The details consist of the PAN holder's name, PAN holder's address, and PAN holder's phone number.

API Call for search by PAN: <http://localhost:3000/api/byPan/123456789>



The screenshot displays a web interface for searching by PAN number. At the top left, the header identifies the organization as the Government of Nepal, Ministry of Finance, and Financial Controller General Office (FCGO). The main content area features a search form with four input fields and a search button. The first field, labeled 'PAN Number*', contains the value '123456789'. The second field, 'PAN holder's Name', contains 'Ram Sharma'. The third field, 'PAN holder's Address', contains 'Lekbeshi, Surkhet'. The fourth field, 'PAN holder's Phone Number', contains '9841000001'. A blue 'Search' button is positioned to the right of the PAN Number field.

Figure 3.23: Search by PAN page

3.4.8 Developing Analysis Page

The analysis page was developed by using ChartJS. ChartJS is an external library that is used to design and develop different kinds of charts. It supports a variety of charts like line charts, bar graphs, pie charts, etc. The data for the analysis was taken from the database with its own distinct API call. The response of the API call contained custom formatted data, which is very suitable for creating charts. This also added an increased layer to the security of the system.

The analysis was done using the data stored in the database. The main purpose of this page was to have a quick breakdown of the revenue collected and the revenue left to be collected. It shows the breakdown using different graphs. The data is loaded using the API call. A separate API route /analysis gives only the required data to improve security. The different data that comes from the API call are as follows:

API Call for analysis data: <http://localhost:3000/api/analysis>

```

  [
    {
      "GlobalBankLtd": 51800,
      "NepalRastriyaBank": 78929,
      "NabilBankLtd": 22200,
      "NepalBankLtd": 28520,
      "paid": 78920,
      "unpaid": 114329,
      "seventythree": 42000,
      "seventyfour": 14100,
      "seventyfive": 23000,
      "seventysix": 17090,
      "seventyseven": 9720,
      "seventyeight": 43039,
      "seventynine": 63300
    }
  ]

```

Figure 3.24: Response from API call

The API gives the necessary data needed for the analysis. The response contains the data such as the total amount collected per bank, per fiscal year, and the total amount paid as well as the amount left to be paid. All these data are then used in the different axis for the charts to be shown.

For the Bar Graph of paid and unpaid, the paid data and unpaid data from the API call was taken and then stacked together against the total amount.

For the line chart, the data regarding the total amount collected per bank is taken from the API and then plotted regarding the individual bank against the total amount collected.

For the pie chart, the total amount collected per year was taken from the API. Then the data was displayed as a sum broken down into years in a pie chart.

Code Snippet to create the polar chart using Data from the API and ChartJS:

```

const polarChart= new Chart ("polarChart", {
  type: 'pie',
  data: {
    labels: ['2073', '2074', '2075', '2076', '2077', '2078', '2079'],
    datasets: [{

```

```

label: 'Amount according to year (2073-2079)',
data: [this.sevThree, this.sevFour, this.sevFive, this.sevSix, this.sevSev,
this.sevEight, this.sevNine],
backgroundColor: [
'rgb (255, 99, 132)',
'rgb (75, 192, 192)',
'rgb (255, 205, 86)',
'rgb (201, 203, 207)',
'rgb (54, 162, 235)',
'rgb (255, 99, 132)',
'rgb (175, 12, 92)'
],
hoverOffset: 6
}}
}
})

```



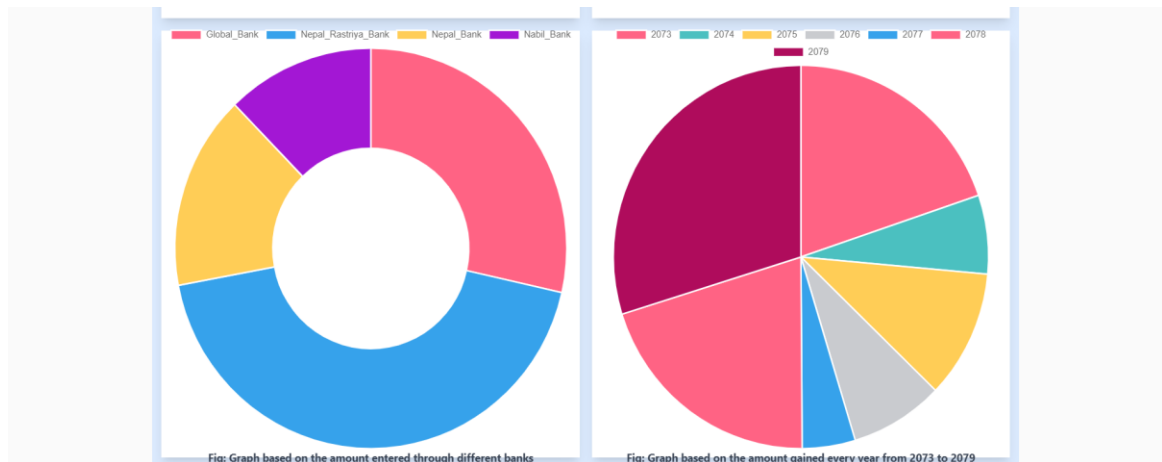


Figure 3.25: Analysis Page

3.4.9 Testing

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and performed allocated functions. The testing process is carried out to make sure that the product exactly does the same thing it is supposed to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are to achieve: -

- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.

Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. In this movie recommendation system, the individual units are individually tested by providing one or a few inputs and expected for a single output.

Table 3.2: Unit Testing on Reading Payment Status Data from API

Test ID	Description	Action	Expected Result	Actual Result	Pass/Fail
1	Reading of Paid Pan Data	Navigate to Paid voucher	A table containing all the paid vouchers should be shown	A table containing all the paid vouchers was shown	PASS
2	Reading of Unpaid PAN data	Navigate to unpaid voucher	A table containing all the unpaid vouchers should be shown	A table containing all the unpaid vouchers was shown	PASS

Table 3.3: Unit Testing on Search by PAN Function

Test ID	Description	Action	Expected Result	Actual Result	Pass/Fail
1	Searching and listing user data by PAN	Input a PAN number on the PAN search page	Users' details such as Name, Address and phone number of the input PAN should be shown	Users' details such as Name, Address and phone number of the input PAN is shown	PASS

Table 3.4: Unit Testing on Create Receipt Function

Test ID	Description	Action	Expected Result	Actual Result	Pass/Fail
1	Data entered should be stored	Inputting user data in create receipt page	The input form data on the create receipt page should be stored in the database	The input form data on the create receipt page is stored in the database	PASS

CHAPTER 4: CONCLUSION AND LEARNING OUTCOMES

4.1 Conclusion

The experience working on this project was outstanding. The project met its purposed objectives. The project was a proof of concept and now it is slowly being integrated into the production environment. It is built using modern technologies and is extremely scalable and flexible. It has lower external code dependencies and uses a robust architecture to handle all the requests. It was a very rewarding experience and provided quality professional as well as technical exposure.

4.2 Learning Outcome

I learned a lot about frontend development and Angular after completing the three-month internship at SimplySoft. The internship program gave me the opportunity to grow professionally while also introducing me to a real working environment. Below are some of the main lessons I learned during my internship:

I learned about working in a fast-paced agile environment. I also learned about the importance of client interaction and how to understand their needs. I understood the process of converting ideas into reality by utilizing effective design mechanisms. I gained the knowledge of building proper scalable applications by using the right tools and frameworks. I learned about creating charts using ChartJS to provide valuable insights as well as designing and implementing a REST API.

I got to know about the different standards and procedures that were followed before the code was even written. I also gained valuable insights about dealing with such high value clients. I got to experience the industry standard for which SimplySoft was known for. I got learned about the best practices used to build a modern scalable web application.

References

(SimplySoft 2021). *About SimplySoft Technologies*. From <http://simplysoft.technology/>

(Sufyan bin Uzayr, 2019) Sufyan bin Uzayr, Nicholas Cloud, Tim Ambler 2019
JavaScript Frameworks for Modern Web Development: The Essential Frameworks, Libraries, and Tools to Learn Right Now.

Appendices

Government of Nepal
Ministry of Finance
Financial Controller General Office (FCGO)

New Revenue Payment of Federal Government

bho

Code	District Name	Select	Code	Office Name	Select
1	Bhojpur	<input type="radio"/>			

< Previous 1 Next >

```
<div
  *ngIf="
    formName.controls.districtName.valid &&
    formName.controls.officeName.valid
  "
  class="w-full flex justify-end"
>
```

```
<button class="p-2 m-2 bg-blue-500 rounded" (click)="onClick(formName)">
  Next
</button>
</div>
<div class="flex flex-row w-full h-auto" [formGroup]="formName">
  <table class="w-full">
    <thead>
      <tr>
        <th>Code</th>
        <th>District Name</th>
        <th>Select</th>
      </tr>
    </thead>
    <tbody class="divide-y">
      <tr>
        *ngFor="
          let data of sortedListDistrictData
          | paginate: { itemsPerPage: 10, currentPage: p };
          let i = index
        "
      >
```

```
onClick(formName: any) {
  this.Router.navigateByUrl(
    '/formpage/' +
    this.formName.controls.districtName.value +
    '/' +
    this.formName.controls.officeName.value
  );
  console.log(this.formName.value)
}
```

```
this.service.getLocation().subscribe((e: any) => {
  this.districtdata = e;

  this.province1 = this.districtdata.filter((element: any) => {
    return element.pname == 'province1';
  });

  this.province2 = this.districtdata.filter((element: any) => {
    return element.pname == 'province2';
  });
  this.province3 = this.districtdata.filter((element: any) => {
    return element.pname == 'province3';
  });
  this.province4 = this.districtdata.filter((element: any) => {
    return element.pname == 'province4';
  });
  this.province5 = this.districtdata.filter((element: any) => {
    return element.pname == 'province5';
  });
  this.province6 = this.districtdata.filter((element: any) => {
    return element.pname == 'province6';
  });
  this.province7 = this.districtdata.filter((element: any) => {
    return element.pname == 'province7';
  });
});
```

```

const routes: Routes = [
  { path: '', component: HomepageComponent},
  { path: 'province', component: ProvinceComponent},
  { path: 'pan', component: PanComponent},
  { path: 'paidvoucher', component: PaidVoucherComponent},
  { path: 'unpaidvoucher', component: UnpaidVoucherComponent},
  { path: 'paidpreview/:id', component: PaidpreviewComponent},
  { path: 'unpaidpreview/:id', component: UnpaidpreviewComponent},
  { path: 'startpayment/:id', component: StartpaymentComponent},
  { path: 'federal/:id', component: FederalGovernmentComponent},
  { path: 'formpage/:id/:id', component: FormpageComponent},
  { path: 'analysis', component: AnalysisComponent}
]

```

```

export class MainService {
  reqHeader = new HttpHeaders({
    'Content-Type': 'application/json',
    'Authorization': 'Bearer ' + sessionStorage.getItem('token'),
  });
  private readonly apiUrl: string = 'http://localhost:3000/api';

  constructor(private http: HttpClient) { }

  //fetch location
  getLocation(): Observable<any> {
    return this.http.get<any[]>(`${this.apiUrl}/location`, {
      headers: this.reqHeader,
    });
  }

  //fetch pan using user id
  getPAN(): Observable<any> {
    return this.http.get<any[]>(`${this.apiUrl}/getAll`, {
      headers: this.reqHeader,
    });
  }
}

```