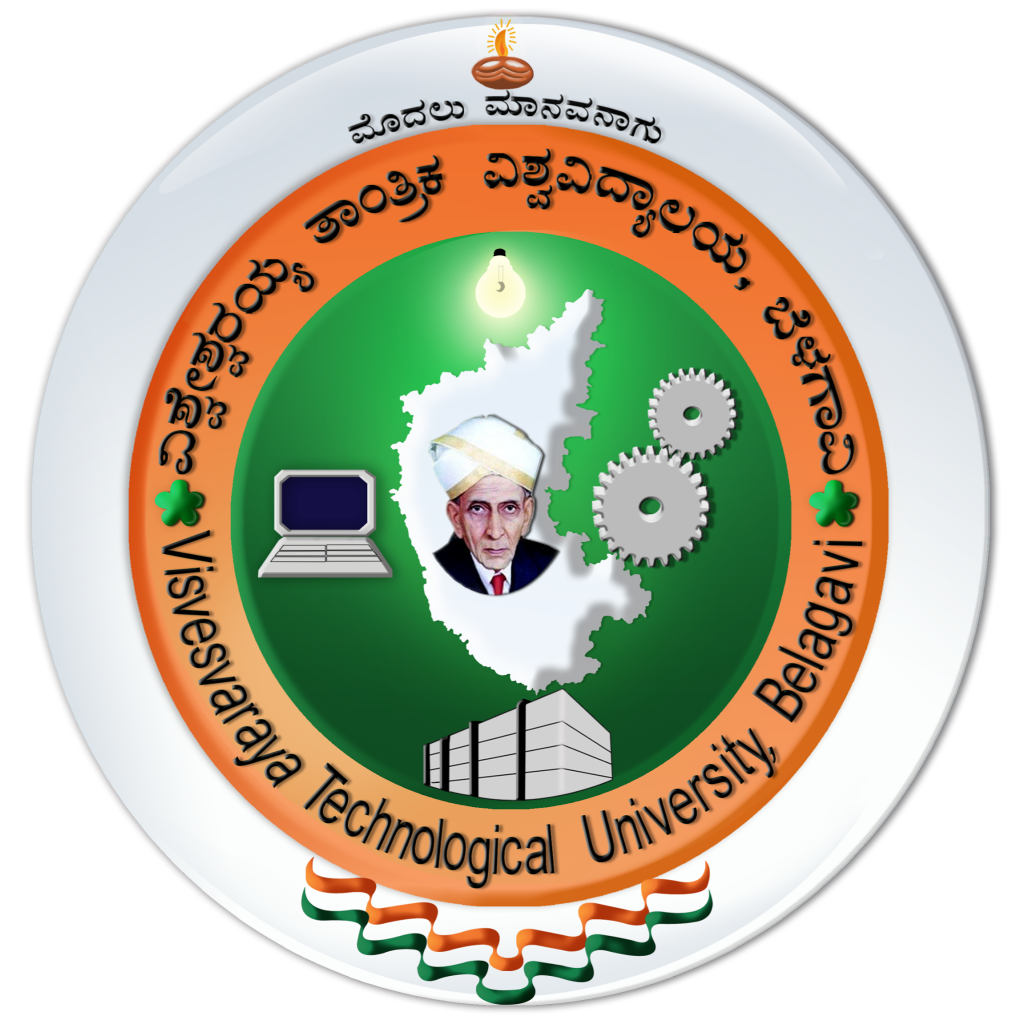
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELGAVI – 590014**



**A MINI PROJECT REPORT ON**

**PAYMENT SYSTEM AND FINANCIAL INTERFACE**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**By**

| **1DB20CS093** | **Sahil Sunil Govekar** |
| --- | --- |
| **1DB20CS091** | **S Suhas** |

**Under the Guidance of**

**Hemanth Kumar N P**

**Assistant Professor**

**Department of CSE, DBIT**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**BENGALURU – 560074, KARNATAKA**

**2022 – 2023**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**BENGALURU – 560074, KARNATAKA**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the Mini Project entitled **“Payment System and Financial Interface”** has been successfully completed by Sahil Sunil Govekar(1DB20CS093) & S Suhas(1DB20CS091) the bonafide students of the **Department of Computer Science & Engineering, Don Bosco Institute of Technology** of the **Visvesvaraya Technological University, Belagavi – 590014**, during the year 2022–2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirements in respect of the Mini Project work prescribed for the Bachelor of Engineering Degree.



**Hemanth Kumar N P Dr. K B Shiva Kumar**

**Mini Project Guide HOD CSE**

**External Viva**

**Name of the Examiners Signature with Date**



**DON BOSCO INSTITUTE OF TECHNOLOGY**

**BENGALURU – 560074, KARNATAKA**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**Declaration**

We, **Sahil Sunil Govekar, S Suhas** hereby declare that the dissertation entitled, **Payment System and Financial Interface** is completed and written by us under the supervision of my guide **Hemanth Kumar N P,** **Assistant Professor, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru**, of the **Visvesvaraya Technological University, Belagavi - 590014,** during the academic year 2022-2023.The dissertation report is original and it has not been submitted for any other degree in any university.

**Sahil Sunil Govekar 1DB20CS093**

**S Suhas 1DB20CS091**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of the people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, with gratitude, we acknowledge all those whose guidance and encouragement served as beacons of light and crowned the effort with success.

The selection of this mini-project works as well as the timely completion is mainly due to the interest and persuasion of our mini-project coordinator Mr. **Hemanth Kumar N P,** Assistant Professor, Department of Computer Science & Engineering. We will remember his contribution forever.

We sincerely thank, **Dr. K B Shiva Kumar**, Professor, and Head, Department of Computer Science & Engineering who has been the constant driving force behind the completion of the project.

We thank our beloved Principal **Dr. B S Nagabhushana,** for his constant help and support throughout.

We are indebted to the **Management of Don Bosco Institute of Technology, Bengaluru** for providing an environment that helped us complete our mini project.

Also, we thank all the teaching and non-teaching staff of the Department of Computer Science & Engineering for the help rendered.

Sahil Sunil Govekar 1DB20CS093

S Suhas 1DB20CS091

**ABSTRACT**

Payment System and Financial Interface is an interface which provides mainly two functionality, end to end payment system that is completely backed using CBDC and loan system which provides based on trustworthy score generated using ML algorithm. Entire interface is designed in such a way that it is completely user friendly interface even a naïve user can use all the features without any difficulty. Every function is just two click process weather is amount transfer or availing loan. Unlike other digital transaction interfaces or gateways (like UPI or Razorpay) which act as a mediator between two banks to process the transaction, our interface is backed using CBDC (Central Bank Backed Digital Currency) is the digital form of currency with RBI’s digital e-Rupee.

This is one of the big steps in establishing a next-generation financial interface. Only few selected country in the world has started working on this technology whereas India has recently startea d pilot program to test and build this interface.

The current version of this interface is capable of performing all major goals of CBDC and other digital payment systems which is running as a web app using flask WSGI (Web Server Gateway Interface) supported by SQL (sqlalchamy) for the database.

As the financial system is considered, security plays a very important role. Moore’s law states that the number of transistors in a microchip increases every two years making it faster and decreasing its cost. This interface is backed with an advanced hashing algorithm with several rounds of salting making it top-notch secure.

**TABLE OF CONTENTS**

| **CHAPTER NO.** | | | | **DESCRIPTIONS** | | **PAGE NO.** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | **DECLARATION** | | **i** | | |
|  | | | | **ACKNOWLEDGEMENT** | | **ii** | | |
|  | | | | **ABSTRACT** | | **iii** | | |
|  | | | | **LIST OF TABLES** | | **iv** | | |
|  | | | | | | | | |
|  |  | | **INTRODUCTION** | |  | | |
|  | **1.1** | | **AIM** | | **1** | | |
|  | **1.2** | | **PROBLEM STATEMENT** | | **1** | | |
|  | **1.3** | | **OBJECTIVES OF THE PROJECT** | | **1** | | |
|  | **1.4** | | **PROPOSED SOLUTION** | | **1** | | |
|  | | | | | | | | |
|  |  | | **SYSTEM DESIGN** | |  | | |
|  | **2.1** | | **SCHEMA DIAGRAM** | | **3** | | |
|  | | | | | |  | | |
|  |  | | **IMPLEMENTATION** | |  | | |
|  | **3.1** | | **LANGUAGE USED FOR IMPLEMENTATION** | | **4** | | |
|  | **3.2** | | **PLATFORM USED FOR IMPLEMENTATION** | | **4** | | |
|  | **3.3** | | **SQL COMMANDS AND QUERIES** | | **5** | | |
|  | **3.4** | | **OUTPUT TESTING** | | **7** | | |
|  |  | |  | |  | | |
|  |  | | **RESULTS** | |  | | |
|  | **4.1** | | **SNAPSHOTS** | | | | **8** |
|  |  | | | | | | |
|  |  | | **CONCLUSION AND FUTURE ENHANCEMENT** | |  | | |
|  | **5.1** | | **CONCLUSION** | | **13** | | |
|  | **5.2** | | **FUTURE ENHANCEMENT** | | **13** | | |
|  | | | | | | | | |
|  |  | **REFERENCES** | | | **14** | | |

**Chapter 1**

**INTRODUCTION**

* 1. **AIM**

To implement interface for payment and loan system which is backed using CBDC and ML algorithm

* 1. **Problem Statement**

There are a few problems in the current digital transaction system

* Dependent on the bank server at both ends to work efficiently while making the transaction
* Limited to a single country or single type of currency, cross country transaction requires a lot of overhead
* Availing of a small-cap loan requires a lot of trustworthiness and background check which is time-consuming
  1. **Objectives of the Project**
* To implement easy to use payment interface that can make payments to any end regardless of respective country and currency
* Awaling and paying back small caps loans with just two simple clicks that include automatic background verification
* The entire interface will have a simple User Interface (UI) so that even comman person without any tech experience can use them
  1. **PROPOSED SOLUTION**

The entire system is backed with CBDC – (Central Bank Backed Digital Currency) where all the transacted amounts exist only in digital form backed using RBI’s digital e-Rupee. This system is usually confused with UPI Interface but there is a major difference. UPI or any other payment interface acts as a mediator between two banks to carry out transactions where the physical currency is stored in banks, whereas this system itself is an interface. This system is a big leap in establishing a **next-generation financial system for India**.

**Payment System**

* Transfer of funds using username, phone number, QR code and Bank account with just two simple clicks
* Each transaction is helpful in building a trustworthy score

**Cross-Border Transaction**

* The amount can be added to the wallet from all other currencies with real-time exchange rate which is achieved using API
* Similar to a naive transaction once the top-up is done

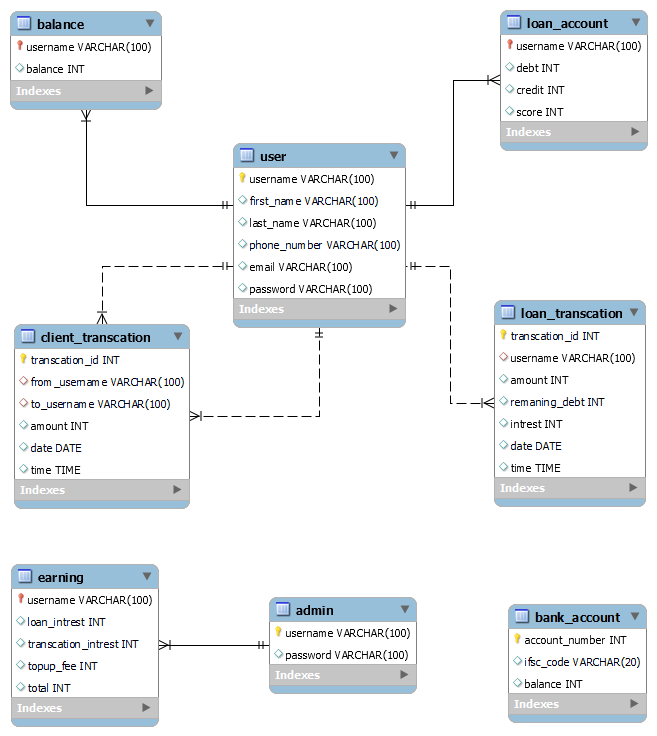
**Loan-System**

* Avail and payback off the loan in two simple clicks
* The score is generated using a machine learning algorithm by tracking the user’s previous day-to-day transactions and loan avail-payback transaction
* Credit is the upper limit of a grantable loan which is calculated from a score of the user

**Chapter 2**

**SYSTEM DESIGN**

**2.1 Schema Diagram**

****

**Chapter 3**

**IMPLEMENTATION**

**3.1 Languages used for implementation**

**Front-end**

* HTML: Providing Structure to the website
* CSS: Designing the elements to improve User-Interface (UX)
* Javascript and jquery: functionality at the front end

**Back-end**

* Python: Backend functionality and building server
* SQL (sqlalchemy): Connecting and perform CRUD operation on database

**3.2 Platform used for implementation**

This interface is a web application that uses Python Flask for server-side rendering web framework. Flask is based on WSGI (Web Server Gateway Interface) and the Jinja2 template engine. The website can be accessed through localhost at specified port numbers using any browser only for the development environment.

The database is stored in form of a model which can be viewed using any database model viewer. CRUD operation on this database is performed using flask sqlalchamy.

**3.3 SQL Commands and Queries**

**Setting up the databse for usage**

from flask\_sqlalchemy import SQLAlchemy #importing library for working with database

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///database.db'

#url in the quotation describes name and location of the database

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db = SQLAlchemy(app)

**Creating new table**

class User(UserMixin, db.Model):

    \_\_tablename\_\_ = "user"

    id = db.Column(db.Integer, primary\_key=True)

    username = db.Column(db.String(100), unique=True)

    fname = db.Column(db.String(100))

    lname = db.Column(db.String(100))

    phoneno = db.Column(db.String(100))

    email = db.Column(db.String(100))

    password = db.Column(db.String(100))

#same way all tables can be created

**To add a table to the database**

from server import app, db

app.app\_context().push()

db.create\_all()

#need to be executed from terminal since table is created only once

**Sample of adding data to the database**

new\_user = User(

                username = s\_username,

                fname = s\_fname,

                lname = s\_lname,

                phoneno = s\_phoneno,

                email = s\_email,

                password = hash\_and\_salted\_password

            )

  db.session.add(new\_user)

  db.session.commit()

#same way new antry can be added to all other table

**Committing Changes**

It is necessary to commit changes in order to reflect it in the database. Everytime new data is added or existing data is modified database needs to be committed

**Reading from the Database**

current\_balance = Balance.query.filter\_by(username=current\_user.username).first()

#this fetches the first entry with constrain in the database (since username is primary key, it fetched db contains only one entry

#inorder to acces the data from aquired row

Balance = current\_balance.balance

#syntax : (fetchedmodelname.columnname)

#similarly all datas are fetched from database

**Updating datas in Database**

current\_balance = Balance.query.filter\_by(username=current\_user.username).first(

current\_balance.balance -= payed\_amount

db.session.commit()

#similarly all values in the database can be updated

**3.4 Output Testing**

**Example : Checking weather current user exist in database or not at log in**

user = User.query.filter\_by(username=input\_username).first()

if not user:

         flash("this username does not exist, Please Register")

         return redirect(url\_for('login\_or\_sign\_up'))

 elif not check\_password\_hash(user.password, input\_password):

         flash("Password zincorrect, PLease Try Again")

         return redirect(url\_for('login\_or\_sign\_up'))

else:

              login\_user(user)

    return redirect(url\_for('home'))

**Example : Checking criteria and availing loan**

cu\_loan\_acc = lnAct.query.filter\_by(username=current\_user.username).first()

admin = Erng.query.filter\_by(username="admin").first()

if loan\_amount > cu\_loan\_acc.credits:

       flash("Insuffecient Credits")

elif cu\_loan\_acc.score < 450:

       flash("Insuffecient Score")

else:

cu\_loan\_acc.debt += (loan\_amount) + (loan\_amount \* 0.12)

cu\_loan\_acc.credits -= loan\_amount

db.session.commit()

user\_balance.balance += l\_amount

#adding revenue to admin

admn.loan\_intrest += (loan\_amount \*0.12)

admn.total += loan\_amount \*0.12

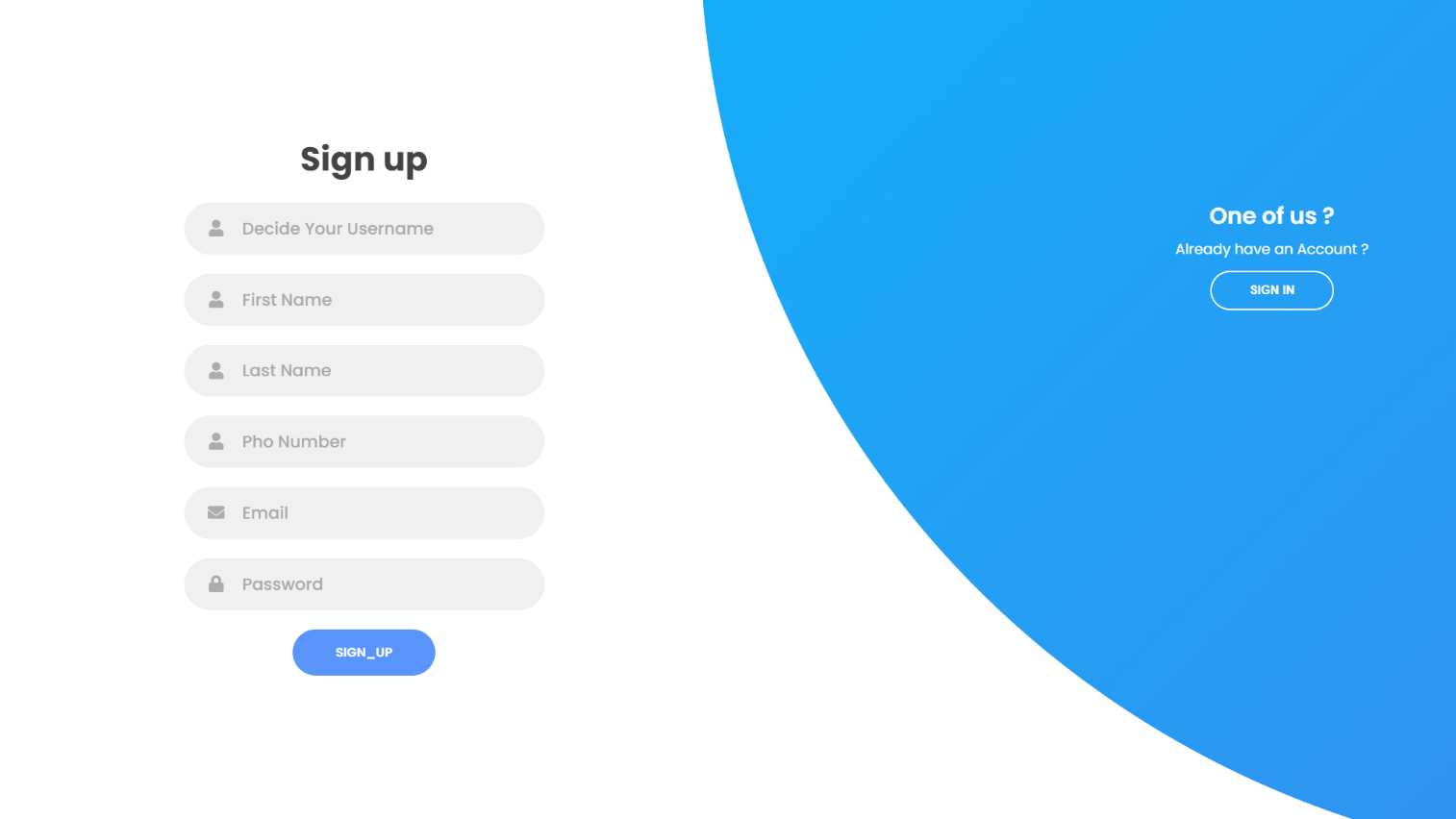
admin\_bal.balance += loan\_amount \*0.12

db.session.commit()

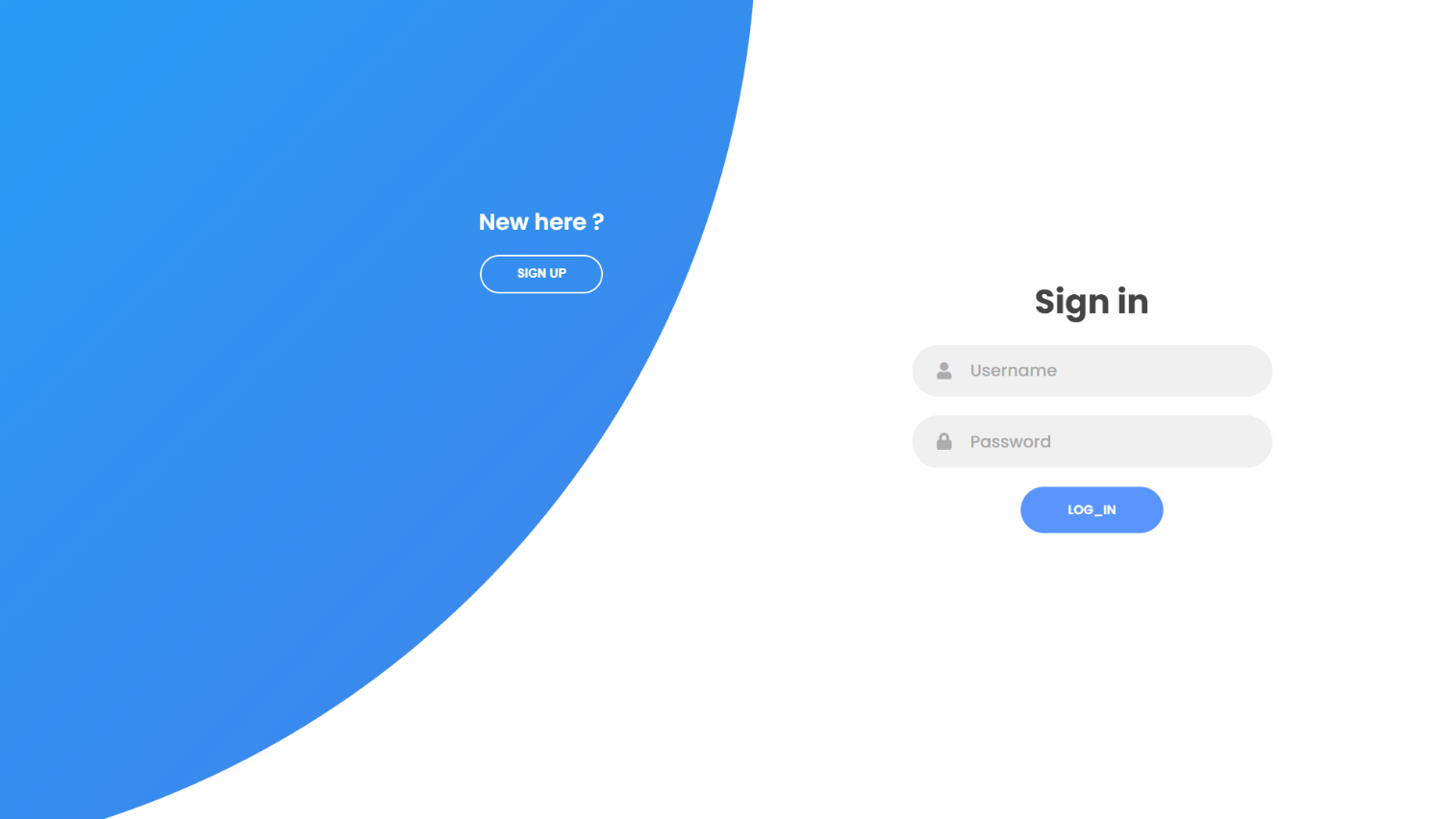
**Chapter 4**

**Results**

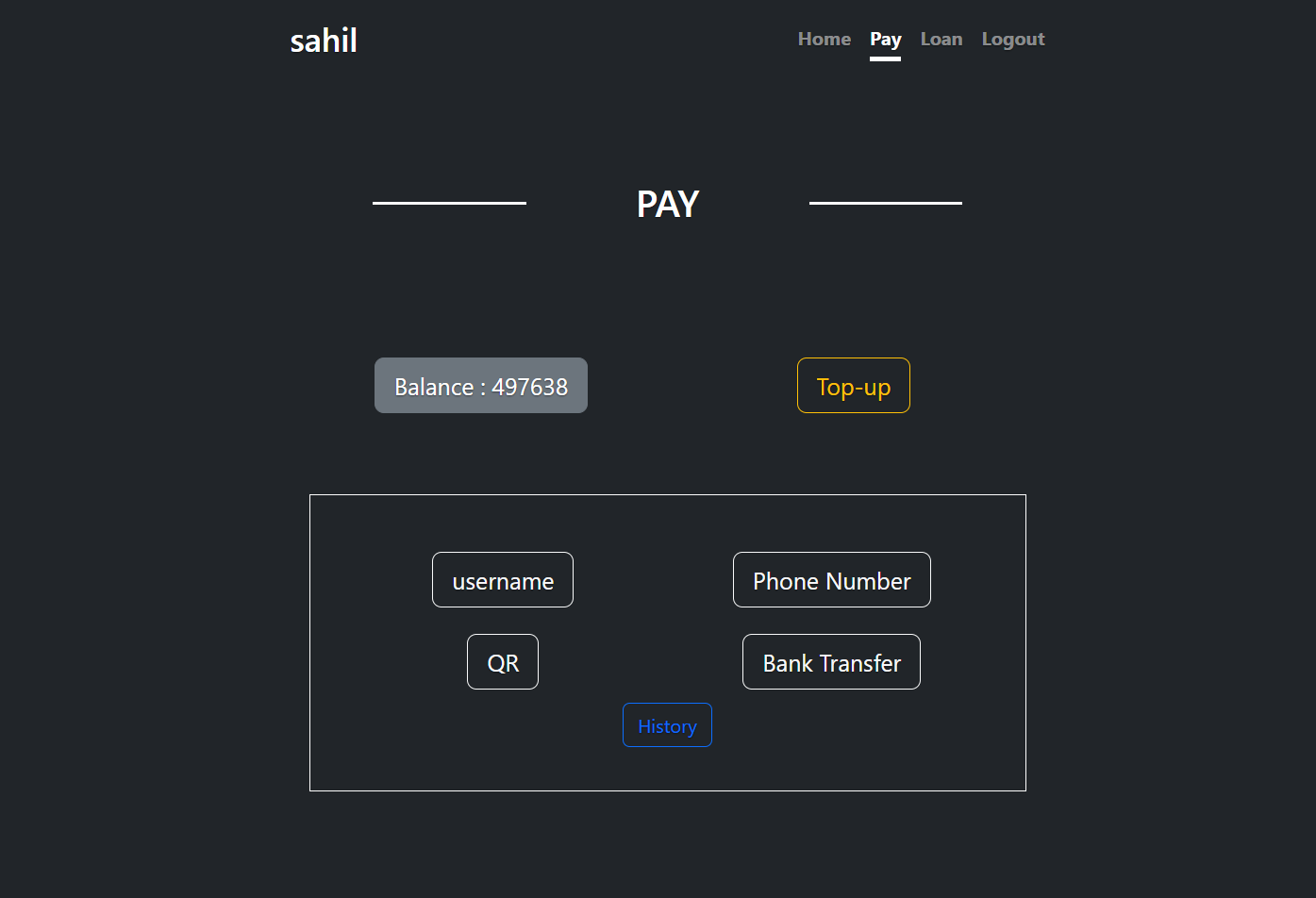
**4.1 Snapshots**



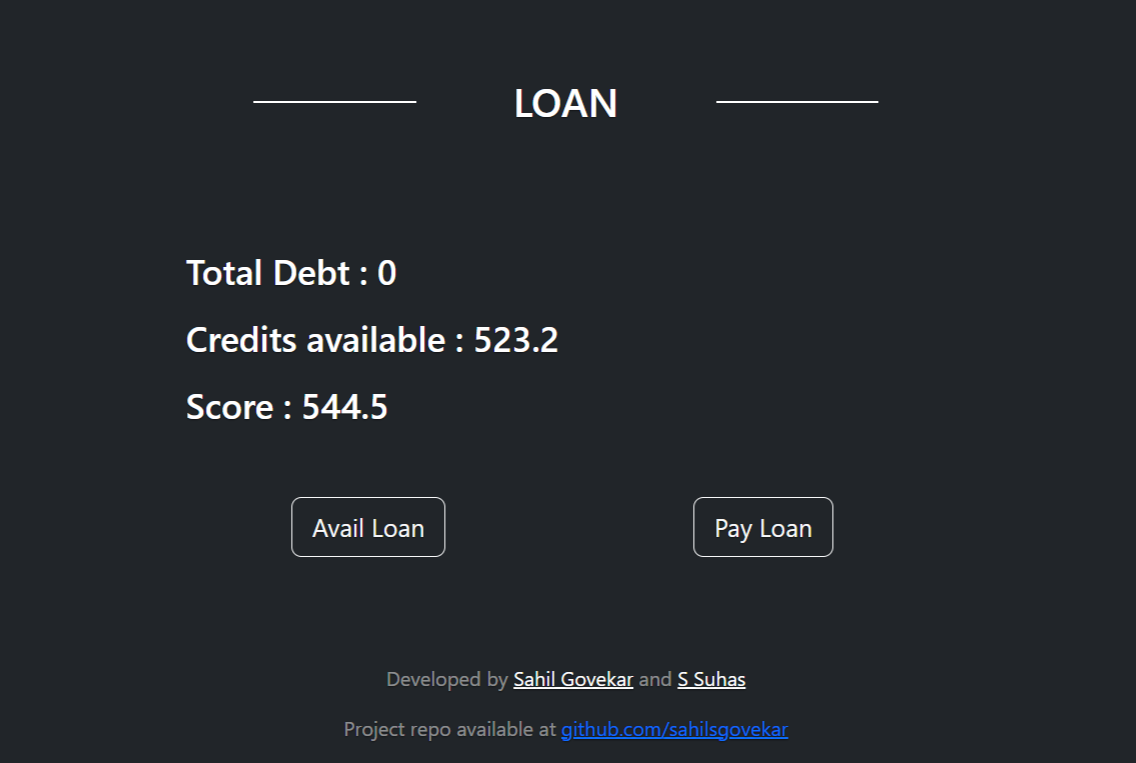
Pic 4.1 Sign up Page

****

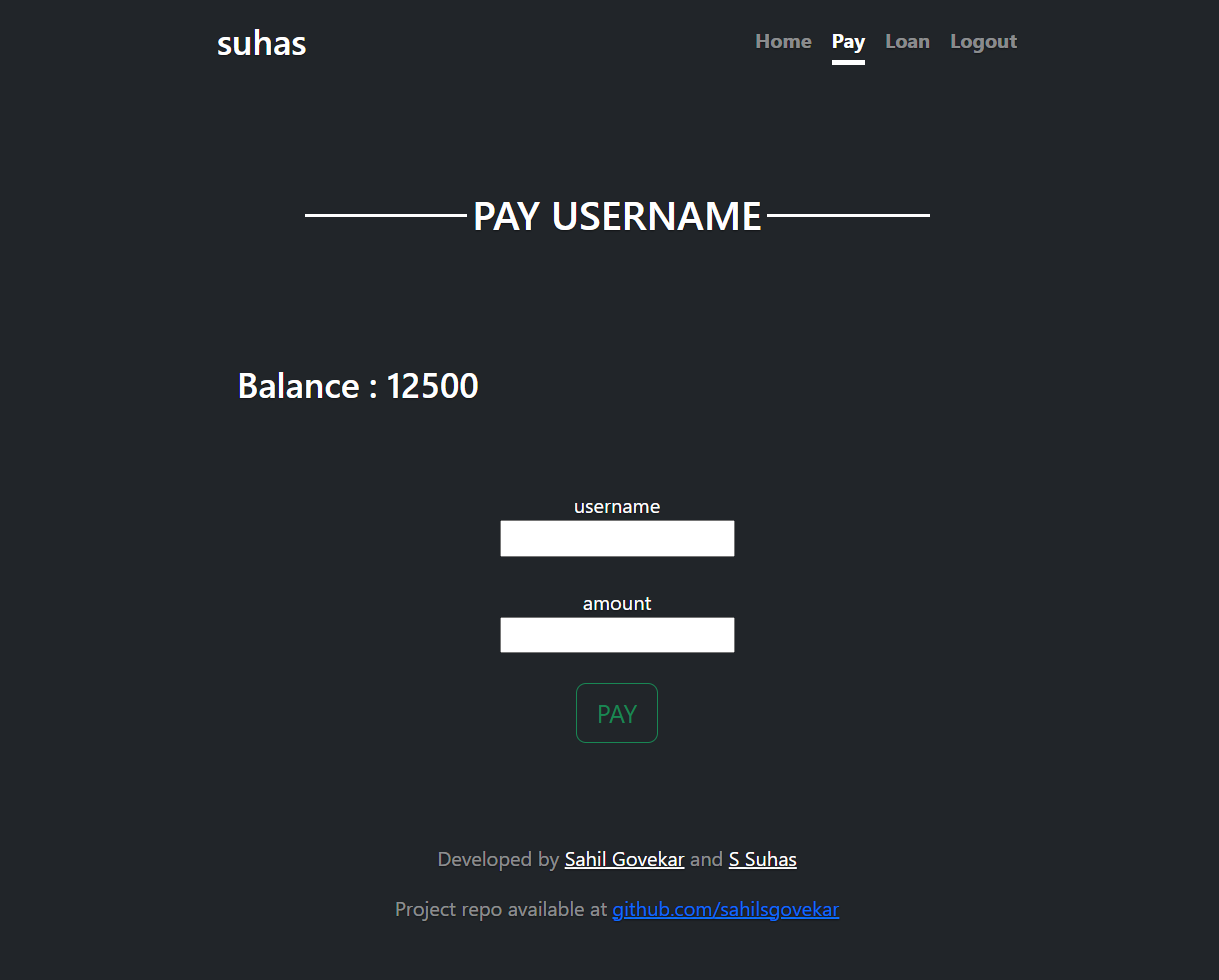
Pic 4.2 Login Page



Pic 4.3 Home Page / Pay

****

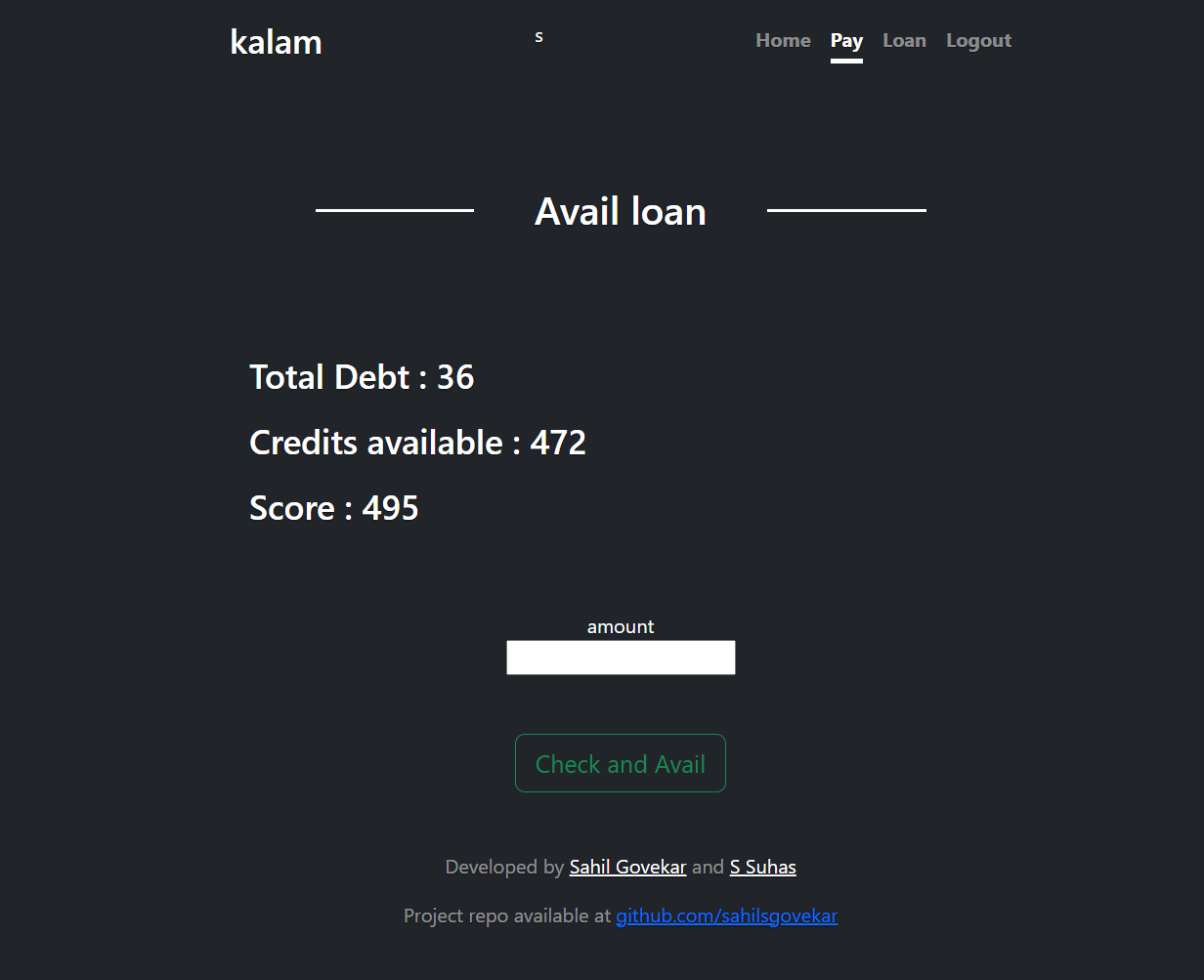
Pic 4.4 Home Page / Loan



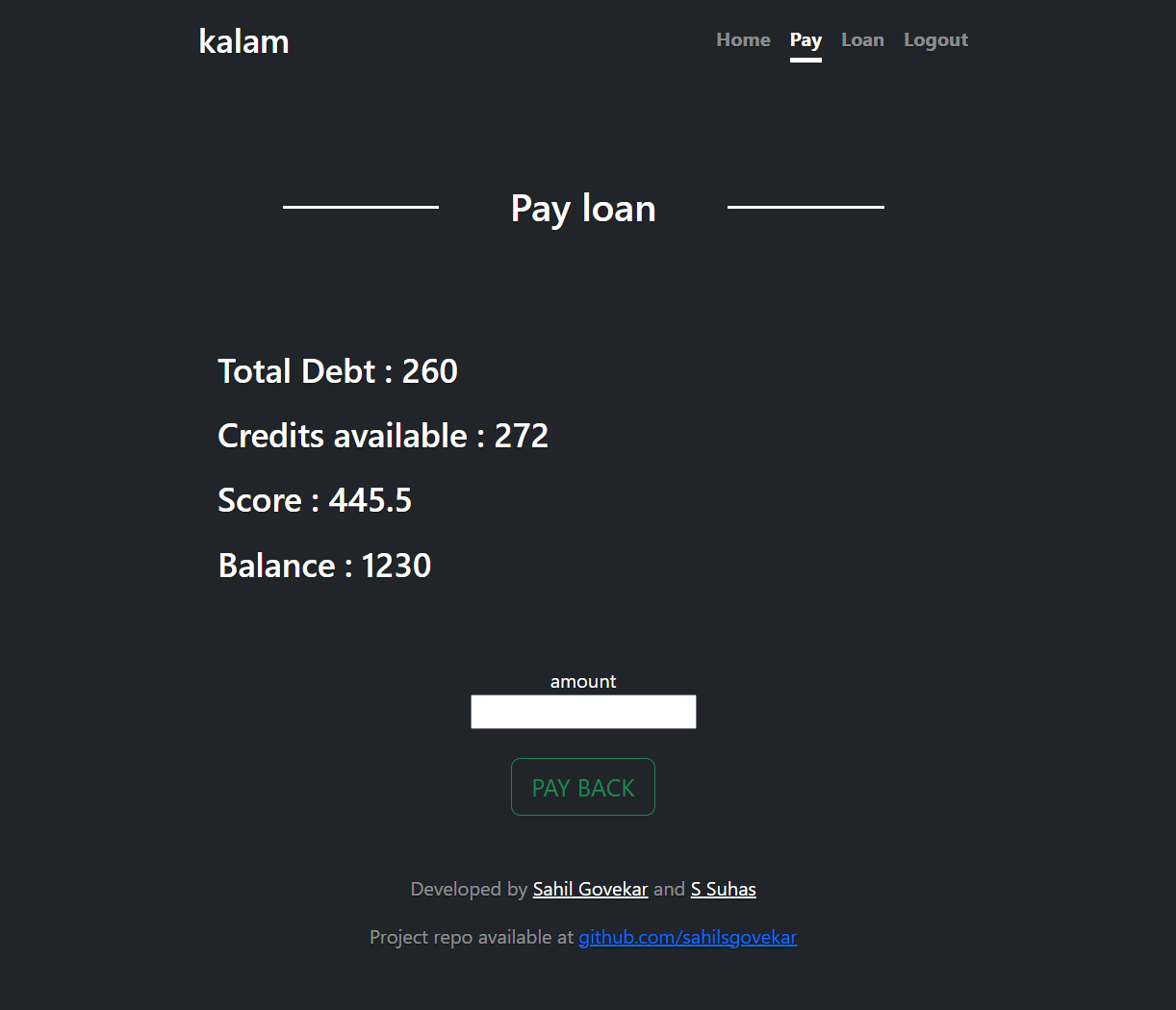
Pic 4.5 Pay Username (Similar for other payment method)



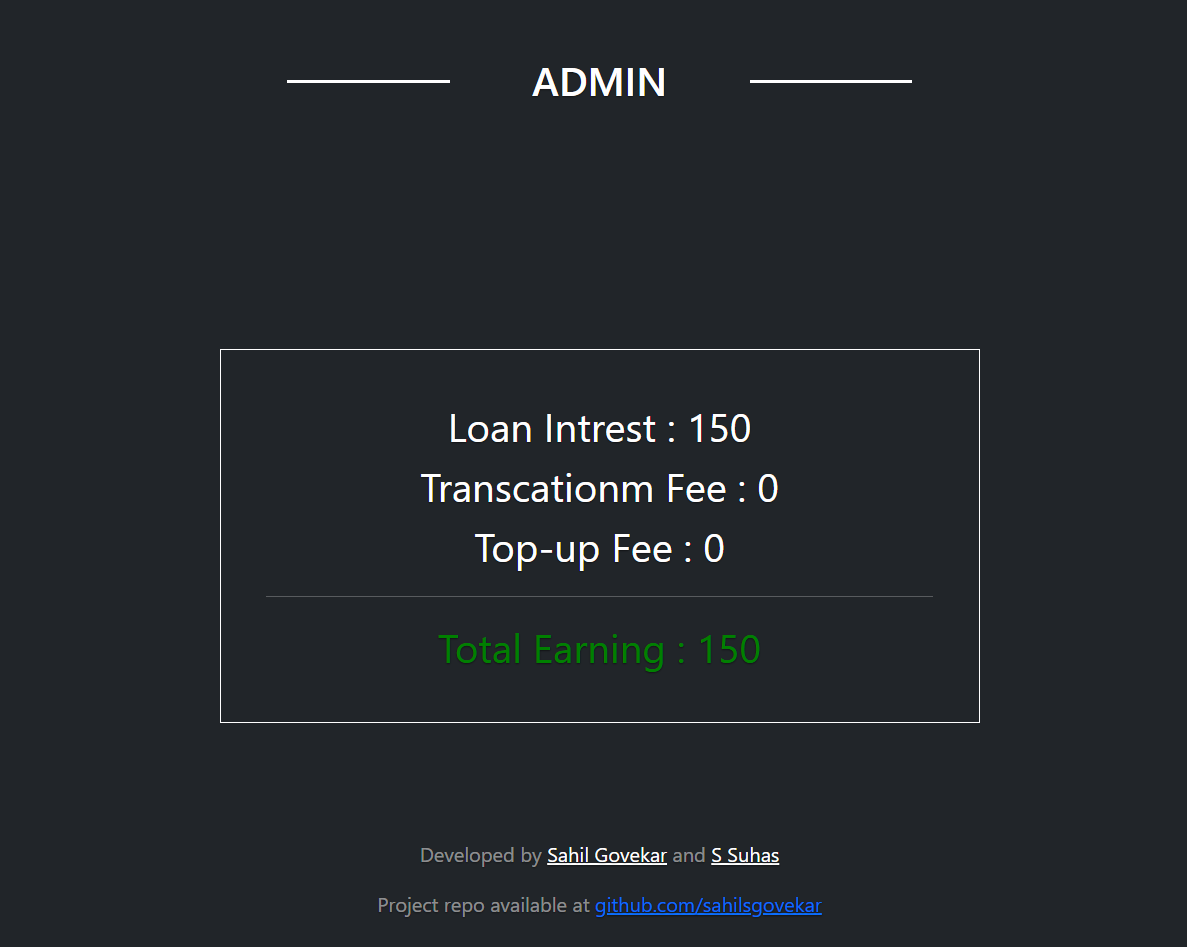
Pic 4.6 Topup wallet



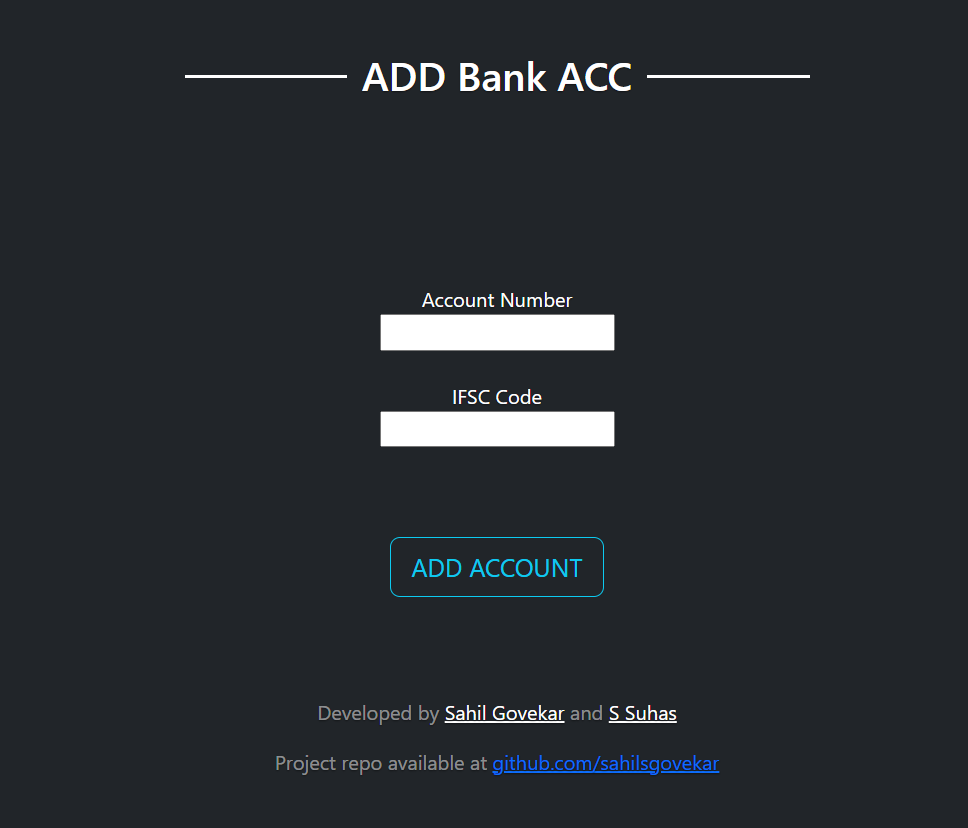
Pic 4.7 Avail Loan



Pic 4.8 Payback Loan



Pic 4.9 Admin Earning (displays total revenue earned from this platform)



Pic 4.10 Adding Bank Account

**Chapter 5**

**Conclusion and Future Enhancement**

* 1. **Conclusion**

This interface is the backbone for establishing a next-generation financial interface, an attempt to build prototype of financial system backed using CBDC and ML algorithms is being achieved.

* 1. **Future Enhancement**

There are several potential enhancements that could be implemented in this interface which can make this future-ready

* The entire system can be converted into decentralized using blockchain making it a robust, user-centric, and ledger-based infrastructure
* Smart contracts can be implemented for every transaction
* The pre-programmed payment system can be implemented which can be used in business while making deals using smart contracts
* A common gateway can be established between several banks across the world to speed up cross-border transaction

**References**

1. Concept note on CBDC by RBI that describes every detail, type and possibilities of this technology

<https://rbidocs.rbi.org.in/rdocs/PublicationReport/Pdfs/CONCEPTNOTEACB531172E0B4DFC9A6E506C2C24FFB6.PDF>

1. GitHub repository of this project

<https://github.com/sahilsgovekar/Payment_System_and_Financial_Interface_using_CBDC>

(further improvement and innovation may reflect here)