**Income Tracker**

**(AY20TECSP60502)**

A **Mini Project Report** Submitted in partial fulfilment of the requirements

of the degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER ENGINEERING**

BY

Balmiki Pooja Rajendra (Roll No 4 )

Anavkar Abhay Chetan (Roll No 2 )

Mhatre Tanvi Sanjay (Roll No 26 )

Supervisor/Guide

\_\_\_\_\_\_\_ Mr. Pravin Jangid\_\_\_\_\_\_

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**SHREE L. R. TIWARI COLLEGE OF ENGINEERING**

**KANAKIA PARK, MIRA ROAD (E), THANE -401 107, MAHARASHTRA.**

**University of Mumbai**

**(AY 2020-21)**

# **Declaration by the Candidate**

I/We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I/We have adequately cited and referenced the original sources. I/We also declare that I/We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I/We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: The 9 May, 2021

**(Balmiki Pooja Rajendra)**

Roll No.: 4 Exam. Seat No.:

**(Anavkar Abhay Chetan)**

Roll No.: 2 Exam. Seat No.:

**(Mhatre Tanvi Sanjay)**

Roll No.: 26 Exam. Seat No.:

|  |
| --- |
|  |
| DEPARTMENT OF COMPUTER ENGINEERING |
| CSM605 Mini Project |
| Sixth Semester, 2020-2021 (Even Semester) |

CERTIFICATE

This is to certify that the **Mini Project** entitled **“Income Tracker”** is a bonafide work of

**Balmiki Pooja Rajendra (Roll No. 4 )**

**Anavkar Abhay Chetan (Roll No. 2 )**

**Mhatre Tanvi Sanjay (Roll No. 26 )**

submitted to the University of Mumbai in partial fulfilment of the requirement of course name “**Mini Project**” having course code **CSM605** for the award of the degree of **“Bachelor of Engineering”** in **“Computer Engineering”**.

**Signature of Supervisor/Guide**

**Name: \_Mr.Pravin Jangid\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of the H.O.D. Signature of the Principal**

**Name: Mrs. Neelam Phadnis Name: Dr. S. Ram Reddy**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
|  |
| DEPARTMENT OF COMPUTER ENGINEERING |
| CSM605 Mini Project |
| Sixth Semester, 2020-2021 (Even Semester) |

# **Mini Project Report Approval**

This mini project report entitled “**Income Tracker*”*** by

**Pooja Balmiki (Roll No. 4 )**

**Abhay Anavkar (Roll No. 2 )**

**Tanvi Mhatre (Roll No. 26 )**

is belonging to the course name “**Mini Project”** having course code **CSM605** submitted as a Term work and approved for the degree of Bachelor of Engineering in Computer Engineering.

**Examiners**

1. Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Internal)

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(External)

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date:**

**Place:**

# **Acknowledgement**

The success and outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project. Project is like a bridge between theoretical and practical work. First of all, we would like to thank the supreme power, the Almighty God, who has always guided us to work on the right path. We record our indebtedness to Shree L. R. Tiwari College of Engineering for giving us an opportunity to initiate our project. We are feeling obliged in taking a moment to sincerely thank Dr. S. Ram Reddy, Principal of the Institute and Prof. NeelamKulkarni, Head of Department of Computer Engineering for the pertinent support and guidance throughout. We would also like to extend a heartfelt thanks to our project guide, Prof. Pravin Jangid to reinforce us. This project could not be completed without the effort and co-operation of all the faculty members of the Department of Computer Engineering. A genuine credit to our parents and families for their abiding encouragement. Last but not the least, we would like to express our gratitude to our friends and respondents for support and willingness to spend some time with us and contribute to the project.

**Balmiki Pooja Rajendra**

Roll No.: 4 Exam. Seat No.:

**Anavkar Abhay Chetan**

Roll No.: 2 Exam. Seat No.:

**Mhatre Tanvi Sanjay**

Roll No.: 26 Exam. Seat No.:

# **Abstract**

In today’s busy and expensive life we are in a great rush to make money. But at the end of the month, we broke off. Every earning person is mostly obsessed at the end of the month as they cannot remember how much they have earned in a particular month. There is no such complete solution present easily or we should say free of cost which enables a person to keep a track of their daily earning. To do so a person has to keep a log in a diary or on a computer, also all the calculations need to be done by the user which may sometimes result in errors leading to miscalculation. Due to the lack of a complete tracking system, there is a constant overload to rely on the daily entry of the income till the end of the month. So we have come over with the idea to track our earnings. We have proposed an native application. This application allows users to maintain a computerized diary to track their daily income. Income Tracker application which will keep a track of Income of a user on a day to day basis. This application keeps a record of your income and will give you a day wise view of your earnings. With the help of this application one can track their daily/weekly/monthly income. This application will also have a feature which will help to track your total earnings on a particular day with the description visible on the screen. This application is capable of being accessed through mobile as well as web browser. A bezier line chart represents the total amount earned on a particular date.

# **Table of Contents**

Income Tracker [i](#_30j0zll)

[(AY20TECSP60502) i](#_1fob9te)

[Declaration by the Candidate ii](#_3znysh7)

[Mini Project Report Approval iv](#_tyjcwt)

[Acknowledgement v](#_3dy6vkm)

[Abstract vi](#_1t3h5sf)

[Table of Contents vii](#_4d34og8)

[List of Figures ix](#_2s8eyo1)

[List of Abbreviations x](#_3rdcrjn)

[1](#_lnxbz9) Introduction 1

[1.1](#_35nkun2) Introduction 1

[1.2](#_1ksv4uv) Background and Motivation 2

[1.3](#_44sinio) Problem statement 3

[1.4](#_2jxsxqh) Project Objectives 3

[1.5](#_3j2qqm3) Project Importance 5

[1.6](#_4i7ojhp) Scope and Limitation of Project Work 6

[1.7](#_2xcytpi) Organization of the Report 6

[2](#_1ci93xb) Literature Review 8

[2.1](#_3whwml4) Survey of Existing System 8

[2.2](#_3as4poj) Problems with Present System 11

[2.3](#_1pxezwc) Limitation existing system or research gap 12

[2.4](#_49x2ik5) Major project Contribution 12

[3](#_147n2zr) Proposed System 13

[3.1](#_3o7alnk) Introduction 13

[3.2](#_23ckvvd) Requirement Analysis and Specification (SRS) 14

[3.3](#_ihv636) Feasibility study report (FSR) 15

[3.3.1](#_32hioqz) Operational Feasibility 16

[3.3.2](#_1hmsyys) Technical Feasibility 17

[3.3.3](#_41mghml) Schedule Feasibility 17

[3.3.4](#_2grqrue) Economic Feasibility 17

[3.3.5](#_vx1227) Legal feasibility 19

[3.4](#_4f1mdlm) Framework 22

[3.5](#_3tbugp1) Details of Hardware & Software 26

[3.6](#_nmf14n) Process Design details 27

[3.6.1](#_111kx3o) Structural UML diagrams 27

3.6.2Behavioral UML diagrams 28

[4](#_206ipza) Methodology 30

[4.1](#_4k668n3) Methodology for solution 30

[4.2](#_2zbgiuw) Implementation Plan 31

[5](#_3ygebqi) Implementation 32

[5.1](#_2dlolyb) Introduction 32

[5.2](#_sqyw64) Implementation of native application 33

[5.3](#_3cqmetx) Implementation of depencies 33

[6](#_4bvk7pj) Result and Discussion 34

[6.1](#_2r0uhxc) Result on Case 1 (PQR) 34

[6.2](#_1664s55) Result of Case 2 (XYZ) 34

[6.3](#_3q5sasy) Result of Case 3 (ABC) 34

[7](#_25b2l0r) Conclusion and Future work. 35

[8](#_kgcv8k) References 36

[9](#_34g0dwd) Annexure – Published Paper 37

# **List of Figures**

[Figure 3-1: Proposed System Framework 24](#_z337ya)

[Figure 3-2: UML Class Diagram](#_qsh70q) 25

[Figure 3-3: UML Sequence Diagram 2](#_2p2csry)6

[Figure 4-1: Agile Methodology](#_1v1yuxt) 27

# **List of Abbreviations**

|  |  |
| --- | --- |
|  |  |
| SRS | Software Requirement Specification |
| VSC | Visual Studio Code |
| FSR | Feasibility Study Report |
| IT | Application programming interfaces |
| ATD | Advanced Technology and Development |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **Introduction**

## **Introduction**

Income management is a topic being very common to people of all ages that are earning from various work now-a-days. In today’s busy and expensive life we are in a great rush to make money and earn from different sources and clients. But at the end of the month, we broke off. Every earning person is mostly obsessed at the end of the month as they cannot remember how much they have earned in a particular month. There is no such complete solution present easily or we should say free of cost which enables a person to keep a track of their daily earning. To do so a person has to keep a log in a diary or on a computer, also all the calculations need to be done by the user which may sometimes result in errors leading to miscalculation. Due to the lack of a complete tracking system, there is a constant overload to rely on the daily entry of the income till the end of the month.

Therefore, this proposed project will help users to track their daily income.

## **Background and Motivation**

Income tracker is a refined system which allows users to efficiently manage his/her income with ease. Tracking income daily can really help users to track the total money earned on a particular day. Once users start off by tracking their income each day, they will be able to get a better idea where and how much they have earned, also users may stay in control and achieve their goal. It will be able to generate your monthly income on a bezier line chart.

1. Great alternative to boost the knowledge in the

field of react.

2. A Great Learning How to work on different

platforms.

3. You can Work from Anywhere and from any

platforms (i.e Android or IOS phone or Desktop).

4.You can work at an Awesome Tech company.

5. There are always Freelance Opportunities.

6. It’s Creative and Fun

## **Problem statement**

Keeping track of income these days is not an easy job for people with busy lifestyles. Making sure each and every income earned is jotted is already a hassle but making sure the right income of being added to the monthly earnings with the source that income is earned is even harder to remember.

So we are developing such a kind of application which will allow users to track their daily earnings.

In this project, the problem statement is ***“To develop a native application that can run on any native platforms named “Income Tracker” for developers and freelancers and also for people earning from various sources so that they can keep track of their daily income using react native.”***

## **Project Objectives**

1. To learn the basics of react native.
2. To study the limitations of the previous expense trackers and develop a more dated system to track income using react native.
3. To develop an improved system in terms of the system being user-friendly.
4. To create a system which has accessibility through mobile as well as web browser.

## **Project Importance**

The main objective of the project is to help the users . It provides an automation procedure of studying the books online. The implementation of this project helps different kinds of users. The users can upload their books to the website by using their unique ID. Also users can purchase course packages for unlimited access of data. Web based Learning is an inexpensive, efficient and comfortable way for users to easily access books and videos and an easier alternative to enhance knowledge and skills.

## **Scope and Limitation of Project Work**

This application can take a good market as it is usable by anyone who is willing to track their income on a daily basis and aims to save for investments and many more. There is not any range criteria or any kind of profession or gender that are focused, it can be used by anyone and anywhere.

The system is divided into one (1) scope which is the user. The scope gives an overview of the functions of the system.

1. User:

- - Enter username and password

- - Input daily income with the description

- - View report i.e total amount on the chart

## **Organization of the Report**

Keeping track of income these days is not an easy job for people with busy lifestyles. Making sure each and every income earned is jotted is already a hassle but making sure the right income of being added to the monthly earnings with the source that income is earned is even harder to remember. Therefore, an easy solution for these issues is to find the right medium for users to maintain a better income tracking diary.

This native application is proposed to help users overcome the struggles they have to endure when they do not track their income or track their income manually or even have troubles with the current existing income and expense tracker that they are using.

The Major Goals Of our project is:

There are certain goals when it comes to development of native application and some of these are to:

* Enhance the quality of the tracking by developing a native platform
* Meet the avaliability of users by making the platform accessible through any device
* Improve the efficiency and effectiveness
* Improve space and time flexibility

# **Literature Review**

## **Survey of Existing System**

Tracking daily income is not so innovative. Many traditional and technological approaches are found to track our expenses and budget with their own functionality. From decades ago and today we have been writing our expenditure in a register to calculate the profit or saving. Not only this, many desktop and mobile applications have been developed for this purpose. Quicken and Microsoft money were the first desktop applications developed decades ago but were not so familiar with the users. Personal capital and dollar bird application were used to visualize the expenses in chart or graphs with the calendar system. QuickBooks were the application for the small business holder to wrap up their whole business. YNAB and Penny were the latest applications which were embedded with AI and applicable for importing expenses automatically. However, Mint was the one which was widely used and trusted.

## **Problems with Present System**

Explaining about the latest application built in this category, YNAB is an expense tracker that gives the automatic tracking of our expenses but no income tracking is done i.e the app takes income as an one input by the user. This application is mobile friendly and is emerging since 2013 but cannot be accessible through any device such as web browsers or IPhones. This application is embedded with AI to define and manage only daily expenses not income. This application does not give any detailed information about our income while our application will track income and shows the information from where and how much users have earned. Moreover, we will be working on the development of the chart showing the income earned on a particular day.

## **Limitation existing system or research gap**

The limitations of the existing system are:

* In the existing system the main module includes the expense tracker by having a fixed input of income of the users i.e no daily income tracking is done on the existing system.
* The existing system has potential limitations in terms of accessibility and system. For the aspect of accessibility, the system is only limited for mobile application. Therefore, users are incapable of accessing the system through a web browser or any other platform.
* Tech Issues.
* Time Consuming.

## **Major project Contribution**

We will be developing such a type of native application which helps users to reduce their effort of remembering their daily income. The app is capable of updating the income through any platform (i.e Android phone, Iphone, Desktop) also the application will have various features of automatically updating the users income on the graph and they can just know the total income they made on a particular day. In short the application will help its users to track their income daily.

# **Proposed System**

## **Introduction**

In this chapter the proposed system Requirement Analysis and Specification(SRS),Feasibility study report (FSR), framework, details of hardware software, structural and behavioural uml diagrams will be discussed for the completion of the whole project.

## **Requirement Analysis and Specification (SRS)**

1. Purpose

The Software Requirements Specification (SRS) will provide a basic description and requirements for the **“Income Tracker”**. Below is a complete understanding of the Income tracker.

In today’s busy and expensive life we are in a great rush to make money. But at the end of the month, we broke off. Every earning person is mostly obsessed at the end of the month as they cannot remember how much they have earned in a particular month. There is no such complete solution present easily or we should say free of cost which enables a person to keep a track of their daily earning. To do so a person has to keep a log in a diary or on a computer, also all the calculations need to be done by the user which may sometimes result in errors leading to miscalculation. Due to the lack of a complete tracking system, there is a constant overload to rely on the daily entry of the income till the end of the month.

Therefore, we are developing such a kind of application that will allow users to track their daily earnings.

1.2 Document Convention

The document is prepared using Microsoft Word and has used the font type ‘Arial’. The fixed font size that has been used to type this document is 12pt with no line spacing. It has used the bold property to set the headings of the document.

1.3 Intended Audience

The intended audience for the project are developers, freelancers, and also the common people earning from different sources

1.4 Project Scope

This application can capture a good market as it is usable by anyone who is willing to track their income on a daily basis and aiming to save for investments and many more. There is not any range criteria or any kind of profession or gender that are focused, it can be used by anyone and anywhere.

2. OVERALL DESCRIPTION

2.1 Product perspective

The main perspective of this project is to provide an online native platform that keeps a record of the income earned on a particular day by users.

2.2 Product functions

The major features are shown below in the UML Sequence Diagram.

2.3 User classes and characteristics

The system is divided into one (1) scope which is the user. The scope gives an overview of the functions of the system.

1. User:

- - Input daily income with the description

- - View report i.e total amount on the chart

2.4 Operating environment

The operating environment is developed in react native i.e native application will help users to reduce their effort of remembering their daily income. The app is capable of updating the income through any platform (i.e Android phone or iPhone or Desktop) also the application will have various features of automatically updating the user’s income on the graph and they can just know the total income they made on a particular day. In short, the application will help its users to track their income daily.

2.5 Design and Implementation constraints

➢ The user can enter a description and earning amount on a particular day.

➢ The entered earning amount is added to the total amount and the chart gets updated as per the amount on a particular day.

2.6 Assumptions and Dependencies

It is assumed that the system developed will work perfectly under Windows OS, and Expo Server, and Expo go app.

3. SYSTEM FEATURES

3.1 Functional requirements

The user has to enter the description and income manually on the income tracker. And then the tracker will display description, income, total income on a particular day, and also the chart gets updated as per the total income on a particular day.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 User interfaces

Users of the system will be provided with the React native user interface.Theuser interface provided by the system is user-friendly.

4.2 Hardware interfaces

Hardware requirement will be the same for both the parties whichare as follows:

Processor-Pentium I or above.

Ram 1GB or above.

Hard disk 300gb

4.3 Software interfaces

The system can be executed on a computer system having any version of the Windows operating system.Software required to make for the working of our project is Operating system: Windows 10, Visual Studio code, Expo.

4.4 Communication protocols and interfaces

The project supports all types of web browsers, Android phones, and iPhones.

5. NON-FUNCTIONAL REQUIREMENTS

5.1 Performance requirements

1. Usability: There is a consistency in all the modules and webpages. To ease the navigation there is a back tab to provide access to the previous page. There is proper instruction on each page.

2. Reliability: Each data record is stored on a well-built efficient database schema. There is no risk of data loss. The internal evaluation of data is well coded.

3. Supportability: The system is well built to support any machine. Maintainability of the system is easy.

4. Performance: In order to ease the accessibility, the types of expenses are categorized along with an option to name on their own. Throughput of the system is increased due to light weight database support.

5. Availability: The system is available all the time, no time constraint.

5.2 Safety requirements

The data entered by the user is safer and cannot be tracked or captured by any third party.

5.3 Security requirements

Since there is only one scope i.e user so, only the user will be able to perform the actions on the build-in environment.

5.4 Software quality attributes

* Availability: The system shall be available 24x7.
* Flexibility: Ability to add new features to the system and handle them conveniently.
* Usability: The system should be able to easily accessible by all users through any native platforms (i.e Android phone or iPhone or Desktop)

6. OTHER REQUIREMENTS

Installation of Expo Go app: For user

## **Feasibility study report (FSR)**

Whenever a project of any kind is to be carried out systematically, through careful consideration and - wherever possible - measurement of the parameters involved, a feasibility study becomes an integral part of the project work. In the field of software development, due mainly to conceptual difficulties in identifying the right parameters to measure, this engineering approach has been slow to take hold. An important question to be taken into account is that a feasibility study should be relatively cheap and quick, and it should inform the decision of whether to go ahead with a more detailed analysis. The input to a feasibility study is an outline description of the system and how it will be used within an organization, and the result should be a report that recommends whether or not it is worth carrying on with the development of the project. A feasibility study involves information assessment (identify information required to answer the three questions), information collection (question information sources to discover the answers), and report writing (make a recommendation about whether or not the system development should continue; propose changes to the scope, budget and schedule; suggest further high-level requirements for the system). The relevant information sources for acquiring the necessary information may include managers of departments where the system will be used, software engineers who are familiar with the type of system that is proposed, technology experts, end-users of the system. One of the cons of the existing system is that the applications are not accessible through all the platforms.

This project is native application which provides a feature for users to track their daily income.

It will permit users to change their income on any platform..

The possible solutions of this project are evaluated and compared by the following criteria: 1) Capable to run the app on different platforms; 2) User-friendly.

After the evaluation of the possible solutions, the most feasible solution for this project is developing the application using react native is selected and the project turns out to be cost-effective, vital and practical.

This project’s purpose is to develop an application to be used on different platforms. The most feasible solution for the project has been chosen and approved and now is ready for further elaboration.

### **3.3.1 Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

This assessment has a simple UI. Anyone with the basic knowledge of android mobile phones and use Income Tracker Application. Income Tracker takes a few seconds approx. 2 seconds to take you from login screen to home page. With a click data are entered.

### **3.3.2 Technical Feasibility**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system.

This assessment focuses on the technical resources available. It helps to determine whether the technical team is capable of converting the ideas into working systems. It also involves evaluation of the hardware, software and other technology requirements of the proposed system.

### **3.3.3 Schedule Feasibility**

This feasibility explains the schedule of the project where the first prototype is completed in four days while the deadline was of five days. At the same time, other tasks were also scheduled where the team members were designing UML diagrams and were carried out in eight days. Further all the designing parts were completed as per schedule which was followed by selecting the appropriate dependencies download. Side by side, the process of documentation was also carried out until the completion of the project.

### **3.3.4 Economic Feasibility**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need for online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.The only cost for building this project is for printing and binding the report files and system uses cost. Additionally, effort and time of every team member is the cost involved for this project. Also, the user does not need to pay a single penny to use this app. Just the use of any platform such as Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP. And hence, Income tracker is economically feasible for any one and on any native platform.

### **3.3.5 Legal feasibility**

It determines whether the proposed system conflicts with legal requirements, e.g., a data processing system must comply with the local data protection regulations and if the proposed venture is acceptable in accordance with the laws of the land.

## **Framework**

Proposed System Framework for Income Tracker

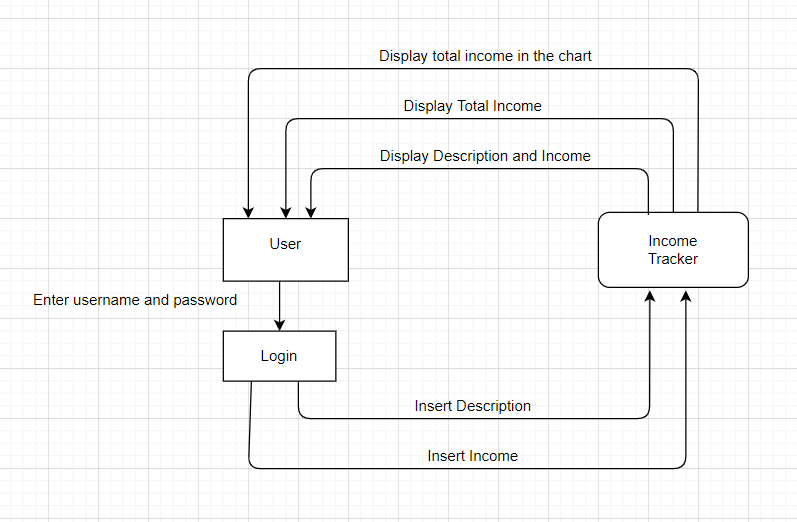


Fig:3-1 Proposed System Framework

## **Details of Hardware & Software**

User interface: React native

Devs Hardware interface:

* Dell Laptop: Processor: Intel Core I3-8th Gen, Memory: 4GB Ram(min),Operating System: Windows
* Galaxy S5 Android Mobile
* IPhone X

Devs Software interfaces:

* IDE: Visual Studio Code
* Browser: Google Chrome
* Expo: To develop, build, deploy and quickly iterate on IOS, Android, and Web apps

## **Process Design details**

Process Design is the act of transformation of an organization’s vision, goals,and available resources into a discernible, measurable means of achieving the organization’s vision. Process design focuses on defining what the organization will do to achieve its financial and other goals.To design refers to the process of originating and developing a plan for products or a service will have an impact on the decision taken during the design of process which produce those products or services vice versa

### **3.6.1 Structural UML diagrams**

Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Class diagram for Income Tracker Application

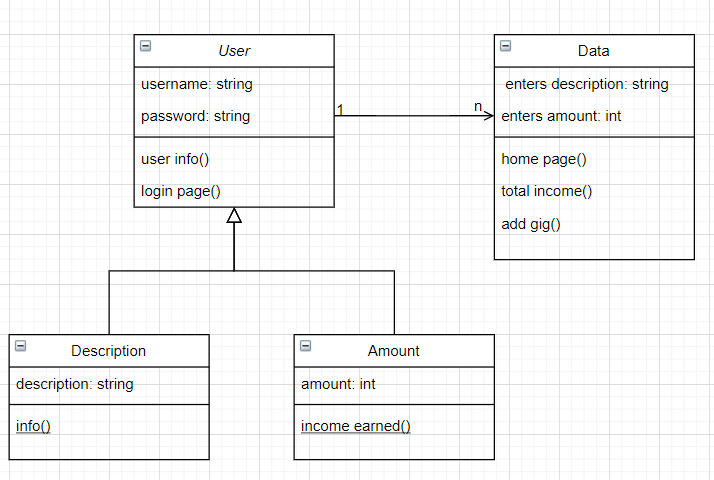


Fig:3-2 UML Class Diagram

### **Behavioral UML diagrams**

Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence diagram for Income Tracker Application

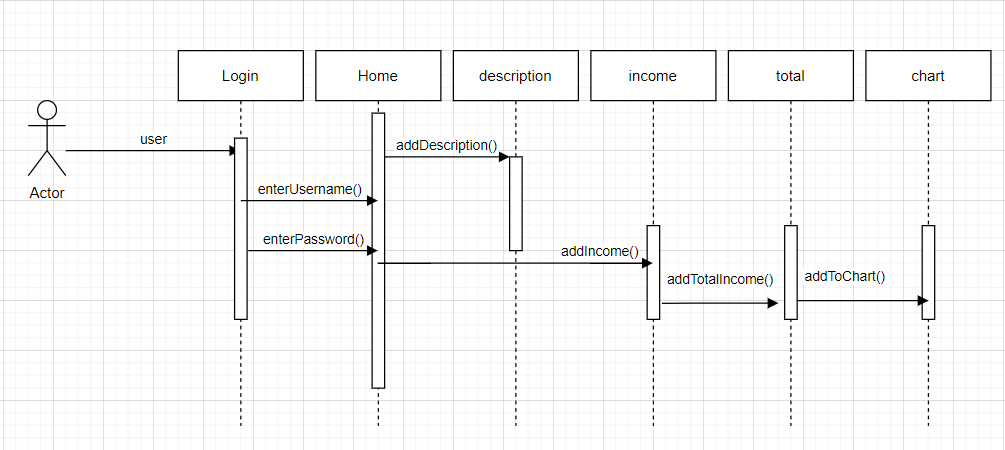


Fig:3-3 UML Sequence Diagram

# **Methodology**

**(Your Step by step technical approach to solve the problem)**

## **Methodology for solution**

Agile development is a user-focused method that takes into consideration user stories in building a system. It helps to make a system more user-friendly in order to satisfy user’s requirements. In addition to its biggest perks as being able to continuously improve, it gives the system the opportunity to adapt to continual changes a system has to undergo in the ongoing changes and upgrades of technologies. The phases of this methodology are as mentioned below:

1. Requirements gathering

2. Design

3. Development

4. Testing

5. Deployment

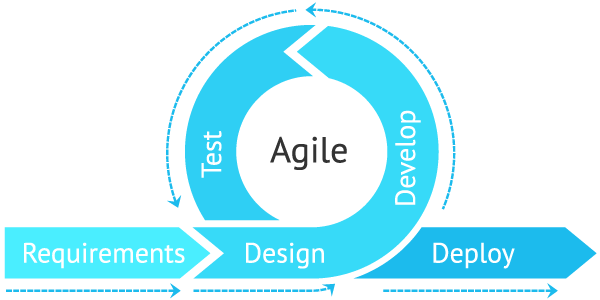


Fig:3-4 Agile Methodology

The steps involved to perform the implementation of the income tracker is listed below:

**1. Requirements Gathering:** The requirements to build up an income tracker are gathered.

**2. Design:** The gathered requirements from the **Requirements Gathering** phase are used in order to identify and to proceed to the designing requirements phase.

**3. Development:** The development of the system begins in this phase.

**4. Testing:** In this phase, the development of the system is at its final stage after the product of the system has gone through a quality assurance test to ensure the system is performing as it should be.

**5. Deployment:** Lastly, in the deployment phase, the system should be performing in great condition given that the system has been tested and is expected to be ready by the end of the last presentation.

## **Implementation Plan**

In this project, the methodology chosen will be discussed for the completion of the whole project. This project will be implementing agile development methodology, as this methodology is open for continuous repetition of processes. The repetition of processes aspect of this methodology will bring great significance to the project for constant amendment.

Implementation process:

1. There are two pages in native application, namely 1. Login, 2. Home.
2. On the login page users can login to the home page by entering username and password.
3. On home page user can perform two functions:
4. Enter the description and the amount he earned on a particular day and the chart will automatically show the data entered and also the total amount will be changed and data user enters will be visible below.
5. Users can return to the login page using the Login button also by clicking the Home arrow.

# **Implementation**

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to develop the application. The system may be totally new replacing an existing manual or automated system or it may be a major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirements.

## **Introduction**

The project is made using react native framework and uses many dependencies namely . The folder structure of the project is as follows:

* App.js
* babel.config.js
* GlobalStyles.js
* HomePage.js
* LoginPage.js
* app.json
* package.json

Here the files named App.js, GlobalStyles.js, HomePage.js, LoginPage.js are the components of our project.

## **Implementation of the native application**

**App.js file**

import React from 'react';

import {SafeAreaView } from 'react-native';

import Homepage from './Homepage';

import LoginPage from './LoginPage';

import { NavigationContainer } from '@react-navigation/native';

import { createStackNavigator } from '@react-navigation/stack';

const Stack = createStackNavigator();

const App = () =>{

return(

    <NavigationContainer>

      <Stack.Navigator initialRouteName='Login'>

        <Stack.Screen name="Home" component={Homepage} />

        <Stack.Screen name="Login" component={LoginPage} options={{

          title:"Sign in to your account"

        }}/>

      </Stack.Navigator>

    </NavigationContainer>

);

}

export default App;

**babel.config.js file**

module.exports = function(api) {

  api.cache(true);

  return {

    presets: ['babel-preset-expo'],

  };

};

**GlobalStyles.js file**

import { StyleSheet, Platform } from 'react-native';

export default StyleSheet.create({

    droidSafeArea: {

        flex: 1,

        // backgroundColor: 'red',

        paddingTop: Platform.OS === 'android' ? 25 : 0

    },

});

**HomePage.js file**

import React, { Component, useState, useEffect } from 'react';

import GlobalStyles from './GlobalStyles';

import { Dimensions,StyleSheet,TouchableOpacity,Text ,View, SafeAreaView,TextInput,Button,Platform } from 'react-native';

import {LineChart} from "react-native-chart-kit";

import dayjs from 'dayjs';

function Homepage({navigation}) {

    var localizedFormat = require('dayjs/plugin/localizedFormat')

    dayjs.extend(localizedFormat)

    const screenWidth = Dimensions.get("window").width;

    const [description,setDescription] = useState('')

    const [amount,setAmount] = useState('')

    const [data,setData] = useState([

      {date:dayjs().format('l'),amount : 499},

      {date:dayjs().subtract(1,'day').format('l'), amount: 2500},

      {date:dayjs().subtract(1,'day').format('l'), amount: 2500},

      {date:dayjs().subtract(2,'day').format('l'),amount: 5500},

      {date:dayjs().subtract(3,'day').format('l'),amount: 3200},

      {date:dayjs().subtract(4,'day').format('l'),amount: 6459},

      {date:dayjs().subtract(5,'day').format('l'),amount: 4500},

      {date:dayjs().subtract(6,'day').format('l'),amount: 7500},

      {date:dayjs().subtract(7,'day').format('l'),amount: 2500},

      {date:dayjs().subtract(8,'day').format('l'),amount: 500},

      {date:dayjs().subtract(8,'day').format('l'),amount: 500},

      {date:dayjs().subtract(9,'day').format('l'),amount: 5000},

      {date:dayjs().subtract(10,'day').format('l'),amount: 4400},

      {date:dayjs().subtract(10,'day').format('l'),amount: 4400},

      {date:dayjs().subtract(11,'day').format('l'),amount: 3200},

      {date:dayjs().subtract(12,'day').format('l'),amount: 2200},

      {date:dayjs().subtract(13,'day').format('l'),amount: 4000},

      {date:dayjs().subtract(14,'day').format('l'),amount: 500},

      {date:dayjs().subtract(15,'day').format('l'),amount: 2000},

    ])

    const [transformedData,setTransformedData] = useState([]);

    useEffect (() =>{

      setTransformedData(transformData(groupBy(data,'date')));

    },[data])

    const groupBy =(array,key) =>

      array.reduce((rv,x) => {

        (rv[x[key]] = rv[x[key]] || []).push(x);

        return rv;

      },{});

    const [total,setTotal] = useState(0)

    const [gigs,setGigs] = useState([{

      description: 'Freelancing',

      amount: 499.99,

      timestamp: dayjs(new Date()).format('l'),

    }

  ])

    const getDates = () => transformedData.map(pair => pair.date);

    const getAmounts = () => transformedData.map(pair =>pair.amount);

    const transformData = (groupedData) => {

      const transformedArray = [];

      Object.entries(groupedData).forEach(entry => {

        const total = entry[1].reduce((total,pair) => total + pair.amount,0)

        transformedArray.push({date:dayjs(entry[0]).format('l'), amount:total})

      })

      const sortedArray = transformedArray.sort((a,b) => dayjs(a['date']).diff(dayjs(b['date'])))

      return sortedArray;

    }

    console.log('Debug',data);

    console.log('The dates',getDates());

    console.log('The amounts',getAmounts());

    console.log('the grouped values are',Object.entries(groupBy(data,'date')));

    console.log('total grouped values are',transformData(groupBy(data,'date')));

    useEffect(() => {

      setTotal(gigs.reduce((total, gig) => total+Number(gig.amount), 0));

    },[gigs])

    const addGig =() =>{

      setGigs([...gigs,{

        description: description,

        amount:Number(amount),

        timestamp:dayjs(new Date()).format('l'),

      }]);

      setData([

        ...data,

        {

        date: dayjs().format('l'),

        amount: Number(amount)

        }

      ]);

      setDescription('');

      setAmount('');

    }

    return (

      <SafeAreaView style={GlobalStyles.droidSafeArea}>

        <View>

          <Text style={{ fontSize: 30, fontWeight: "bold"}}>

            React Native App for Freelancer Devs to Track Income🚀🚀🚀

          </Text>

        </View>

        {Platform.OS === 'ios' ? <TouchableOpacity

            style={styles.loginScreenButton}

            onPress={() => navigation.navigate('Login')}

            underlayColor='#fff'>

            <Text style={styles.loginText}>Login</Text>

          </TouchableOpacity> :

                <Button  title="Login" onPress={() => navigation.navigate('Login')}/>

          }

        <View>

  <Text>Bezier Line Chart</Text>

  <LineChart

  width={screenWidth}

    data={{

      labels: getDates(),

      datasets: [

        {

          data: getAmounts(),

        }

      ]

    }}

    width={Dimensions.get("window").width}

    height={220}

    yAxisLabel="₹"

    yAxisInterval={1}

    chartConfig={{

      backgroundColor: "#e26a00",

      backgroundGradientFrom: "green",

      backgroundGradientTo: "green",

      decimalPlaces: 1,

      color: (opacity = 1) => `rgba(255, 255, 255, ${opacity})`,

      labelColor: (opacity = 1) => `rgba(255, 255, 255, ${opacity})`,

      style: {

        borderRadius: 16

      },

      propsForDots: {

        r: "6",

        strokeWidth: "2",

        stroke: "#ffa726"

      }

    }}

    bezier

    style={{

      marginVertical: 8,

      borderRadius: 16

    }}

    />

  </View>

        <Text style={{fontSize: 20, fontWeight: "bold"}}>Total Income: {total}</Text>

          <TextInput

          style = {{height: 40, padding:20, borderColor: "red",borderWidth:1,marginTop:10,...Platform.select({

            android: {

              height: 60

            },

            ios:{

              height: 60,

            }})}}

          value = {description}

          placeholder="Enter a description"

          onChangeText ={text => setDescription(text)}

          />

          <TextInput

          style = {{height: 40, padding: 20, borderColor: "red",borderWidth:1,marginTop:10,...Platform.select({

            android: {

              height: 60

            },

            ios:{

              height: 60,

            }

          }) }}

          value = {amount}

          placeholder="Enter the amount you made in Rs (₹)"

          keyboardType = 'numeric'

          onChangeText ={text => setAmount(text)}

          />

        <Button disabled={!description && !amount} title='Add Gig🚀' onPress={addGig}/>

        {gigs.map((gig,i)=>(

            <View key={i}>

                <Text >{gig.description}</Text>

              <Text>₹{gig.amount}</Text>

            </View>

        ))}

      </SafeAreaView>

    );

  }

export default Homepage;

const styles = StyleSheet.create({

  loginScreenButton:{

    marginRight:0,

    marginLeft:0,

   marginTop:10,

    paddingTop:5,

    paddingBottom:5,

    backgroundColor:'#007AFF',

    borderRadius:10,

    borderWidth: 1,

    borderColor: '#fff'

  },

  loginText:{

      color:'#fff',

      fontSize:20,

      fontWeight:'bold',

      textAlign:'center',

      paddingLeft : 10,

      paddingRight : 10

  }

  });

**LoginPage.js file**

import React,{useState} from 'react'

import { View, Text,Button,StyleSheet,TouchableOpacity,TextInput,Platform} from 'react-native';

const LoginPage = ({navigation}) => {

    const [username, setUsername] = useState('');

    const [password, setPassword] = useState('');

    const login =() => {

        if(username === 'admin' && password === 'admin'){

            navigation.navigate('Home');

        }

    }

    return (

        <View>

            <Text>It's quick and easy</Text>

            <TextInput

          style = {{height: 40, padding:20, borderColor: "black",borderWidth:1,marginTop:10,...Platform.select({

            android: {

              height: 60

            },

            ios:{

              height: 60

            }

          })}}

          value = {username}

          placeholder="Enter your username"

          onChangeText ={text => setUsername(text)}

          /><TextInput

          style = {{height: 40, padding:20, borderColor: "black",borderWidth:1,marginTop:10,...Platform.select({

            android: {

              height: 60

            },

            ios:{

              height: 60

            }})}}

          value = {password}

          secureTextEntry={true}

          placeholder="Enter your password"

          onChangeText ={text => setPassword(text)}

          />

          {Platform.OS === 'ios' ? <TouchableOpacity

            style={styles.loginScreenButton}

            onPress={login}

            underlayColor='#fff'>

            <Text style={styles.loginText}>Login</Text>

          </TouchableOpacity> :

                <Button  title="Login" onPress={login}/>

          }

        </View>

    )

}

export default LoginPage;

const styles = StyleSheet.create({

  loginScreenButton:{

    marginRight:0,

    marginLeft:0,

   marginTop:10,

    paddingTop:20,

    paddingBottom:20,

    backgroundColor:'#007AFF',

    borderRadius:10,

    borderWidth: 1,

    borderColor: '#fff'

  },

  loginText:{

      color:'#fff',

      fontSize:20,

      fontWeight:'bold',

      textAlign:'center',

      paddingLeft : 10,

      paddingRight : 10

  }

  });

## **Installation of dependencies used in our native application**

**app.json file**

{

  "expo": {

    "name": "income-tracker",

    "slug": "income-tracker",

    "version": "1.0.0",

    "orientation": "portrait",

    "icon": "./assets/icon.png",

    "splash": {

      "image": "./assets/splash.png",

      "resizeMode": "contain",

      "backgroundColor": "#ffffff"

    },

    "updates": {

      "fallbackToCacheTimeout": 0

    },

    "assetBundlePatterns": [

      "\*\*/\*"

    ],

    "ios": {

      "supportsTablet": true

    },

    "android": {

      "adaptiveIcon": {

        "foregroundImage": "./assets/adaptive-icon.png",

        "backgroundColor": "#FFFFFF"

      }

    },

    "web": {

      "favicon": "./assets/favicon.png"

    },

    "description": ""

  }

}

**package.json file**

{

  "main": "node\_modules/expo/AppEntry.js",

  "scripts": {

    "start": "expo start",

    "android": "expo start --android",

    "ios": "expo start --ios",

    "web": "expo start --web",

    "eject": "expo eject"

  },

  "dependencies": {

    "@react-native-community/masked-view": "^0.1.10",

    "@react-navigation/native": "^5.9.4",

    "@react-navigation/stack": "^5.14.4",

    "dayjs": "^1.10.4",

    "expo": "~41.0.1",

    "expo-status-bar": "~1.0.4",

    "moment": "^2.29.1",

    "react": "16.13.1",

    "react-dom": "16.13.1",

    "react-native": "https://github.com/expo/react-native/archive/sdk-41.0.0.tar.gz",

    "react-native-chart-kit": "^6.11.0",

    "react-native-gesture-handler": "^1.10.3",

    "react-native-screens": "^3.1.1",

    "react-native-svg": "^12.1.1",

    "react-native-web": "~0.13.12"

  },

  "devDependencies": {

    "@babel/core": "^7.9.0"

  },

  "private": true

}

# **Result and Discussion**

Steps to be followed in all the native platforms:

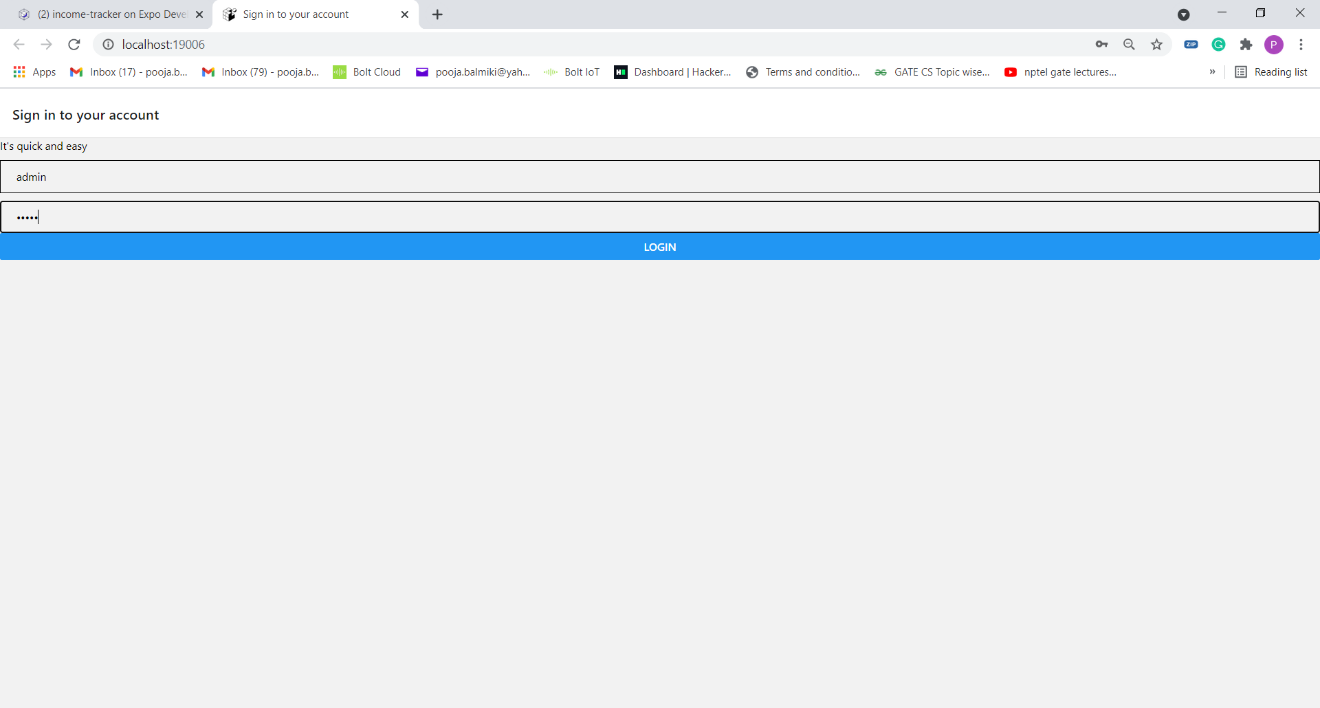
Step1: User enters username and password on LoginPage and then click Login button which directs user to HomePage.

Step2: Users can return to the LoginPage using the Login button also by clicking the Home arrow.

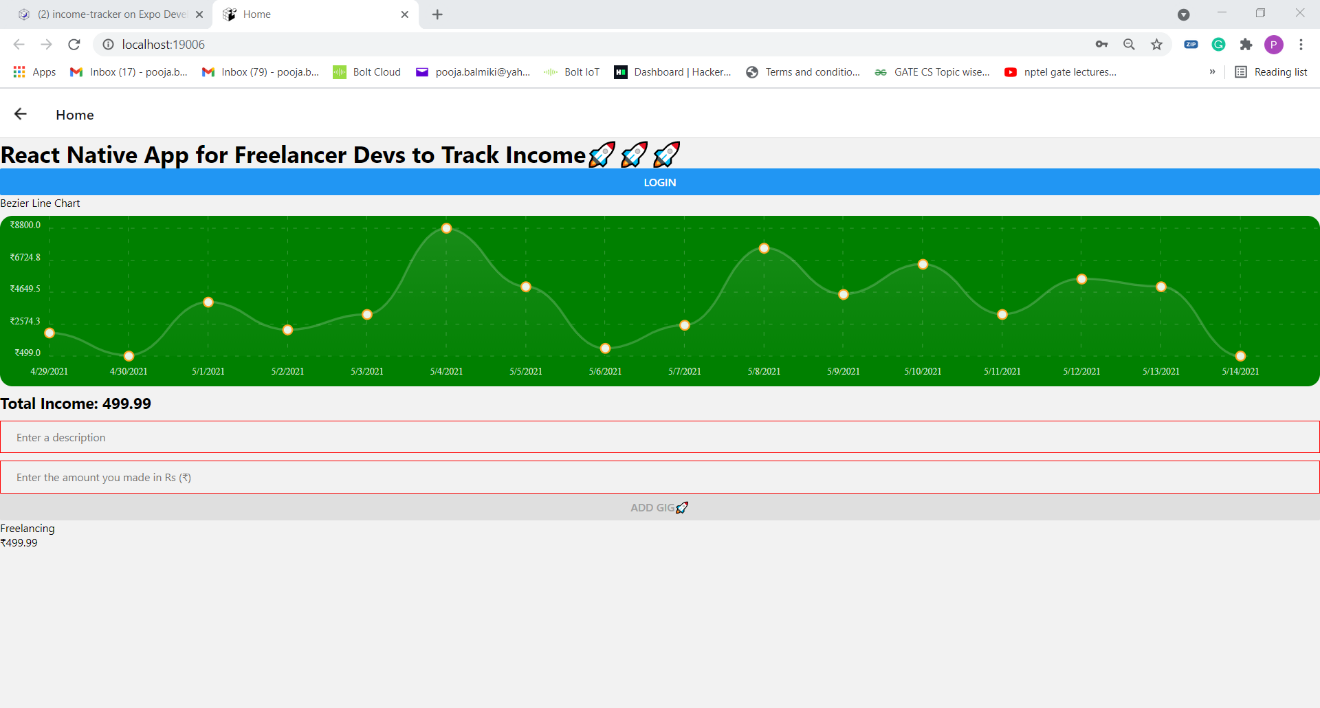
Step3: At Homepage user can enter the description and the amount he earned on a particular day and the chart will automatically show the data entered and also the total amount will be changed and data entered by the user will be visible below.

**6.1 Web Browser View of the application**

LoginPage of Income Tracker



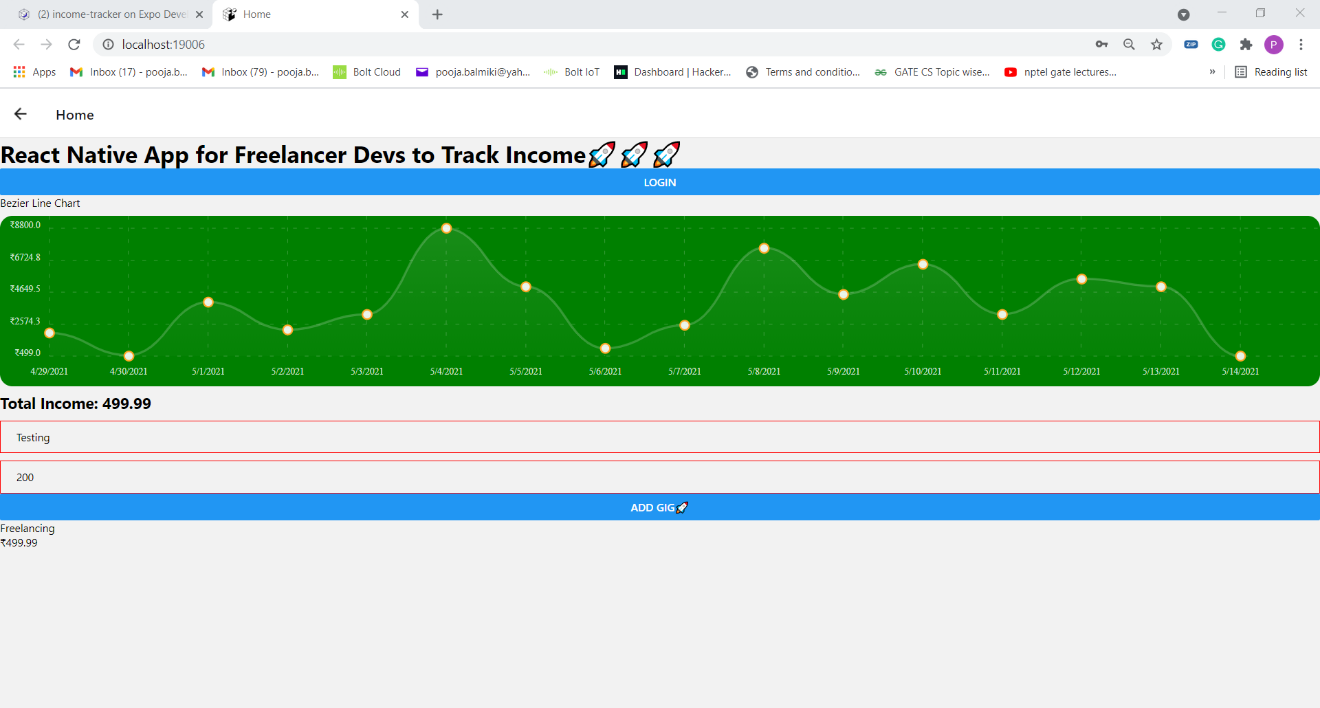
HomePage of Income Tracker



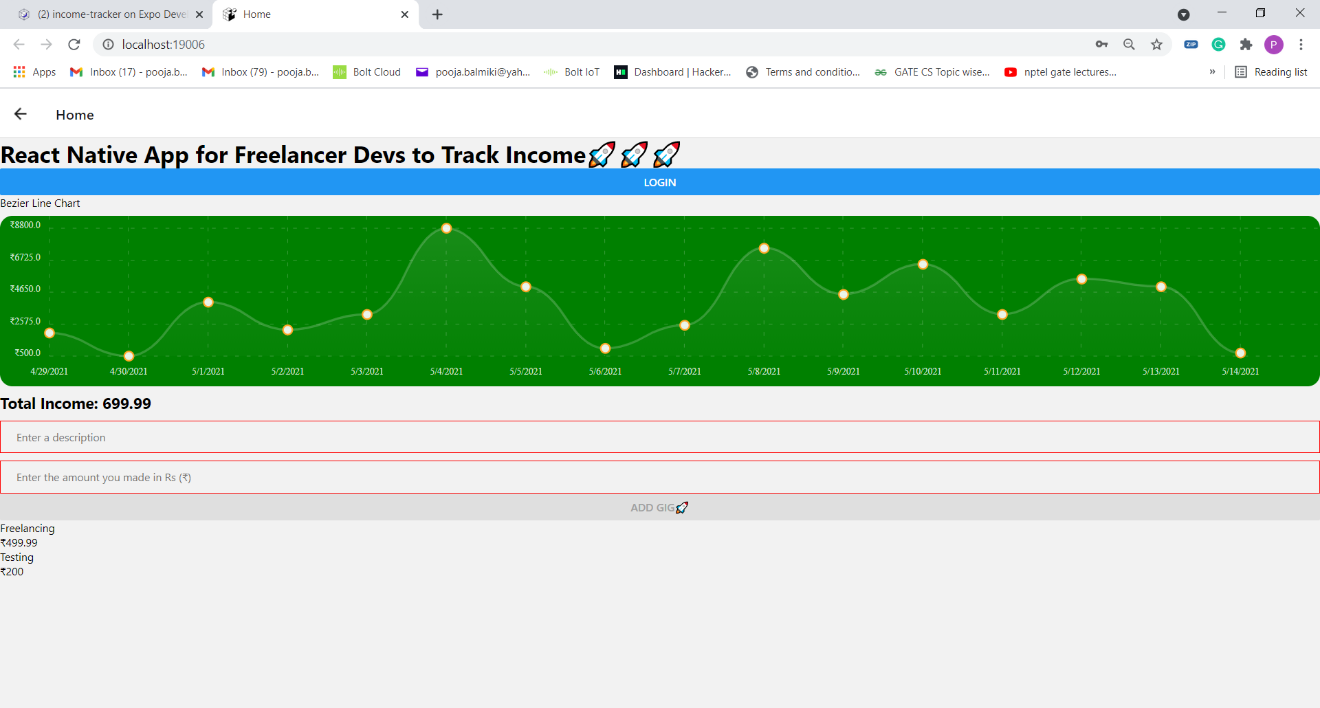
## **.1 Result of Case 1**

User enters the amount 200 earned from Testing.

Before clicking ADD GIG button:

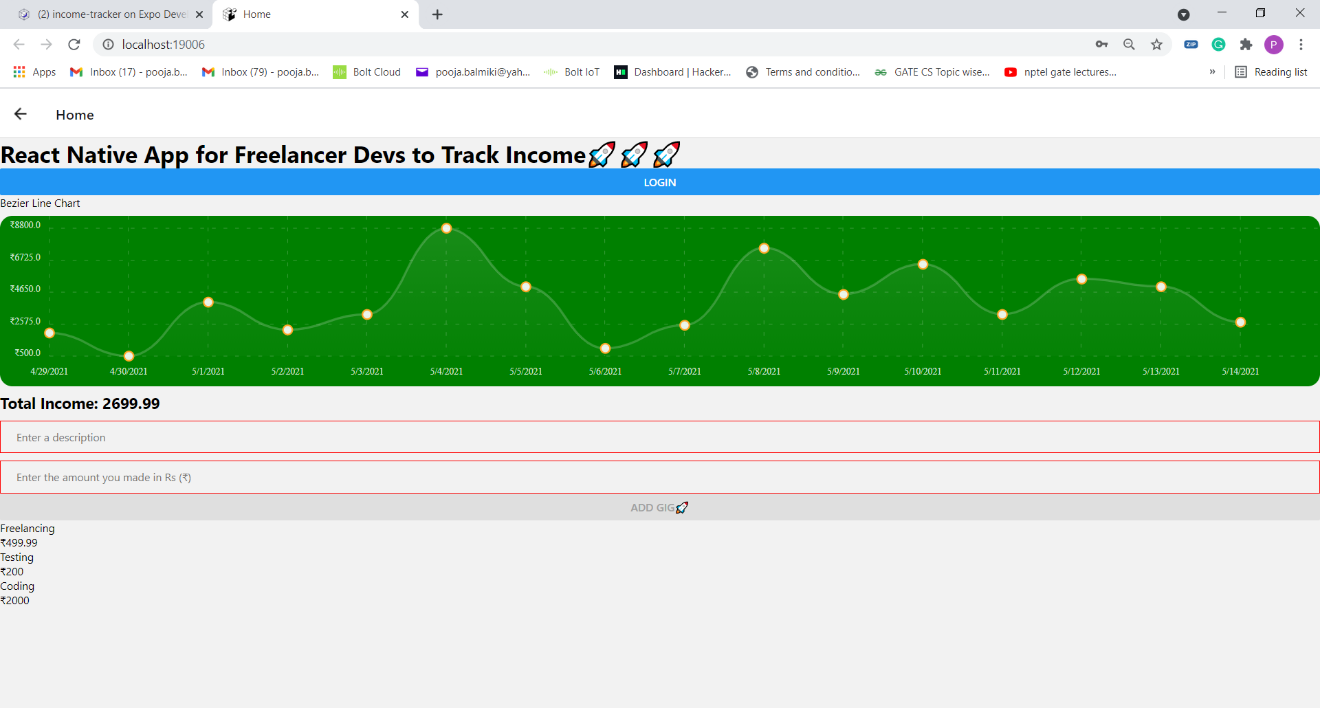


After clicking ADD GIG button



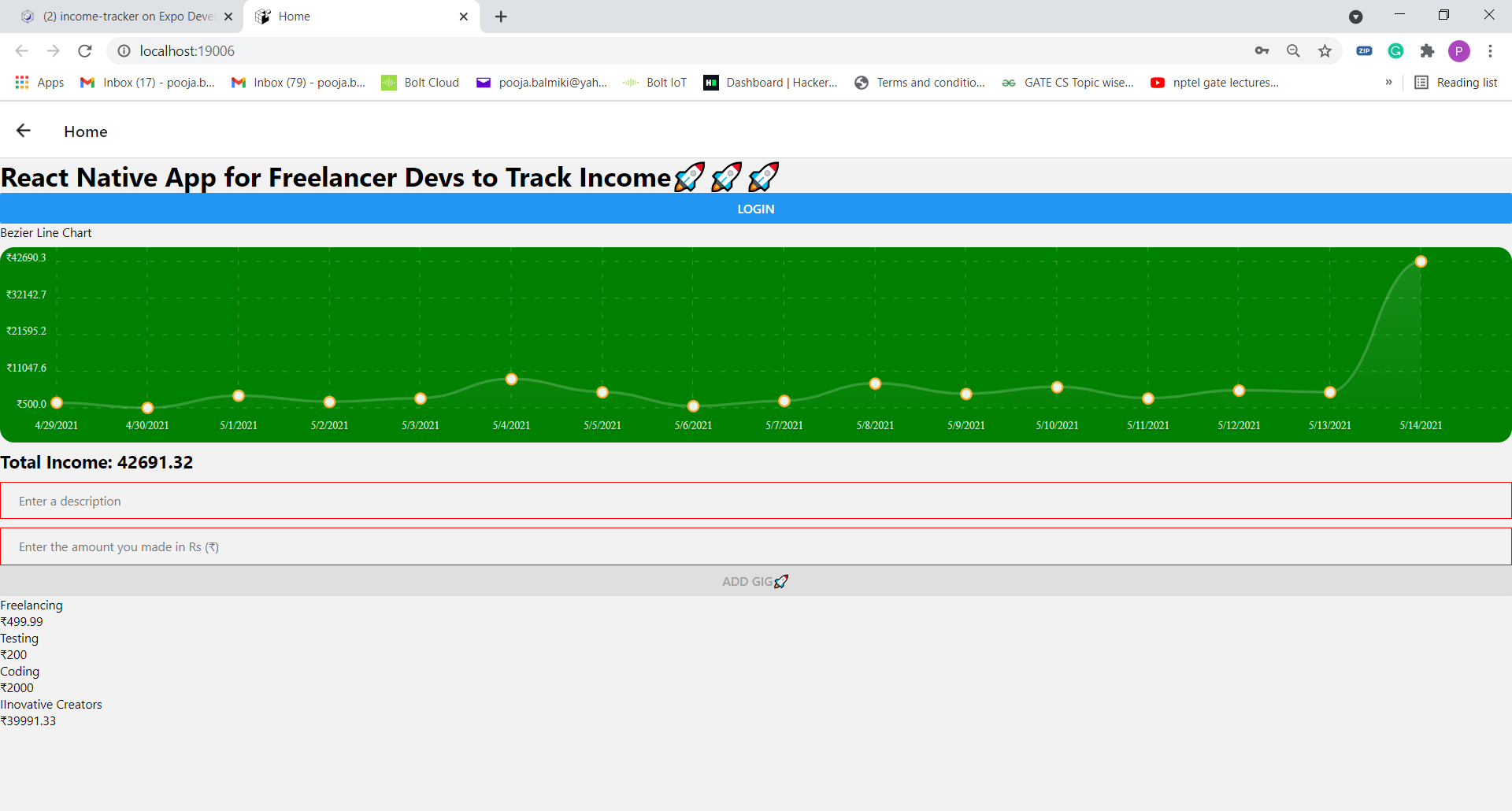
## **6.1.2 Result of Case 2**

User enters the amount 2000 earned from Coding and the final result after clicking ADD GIG button is:



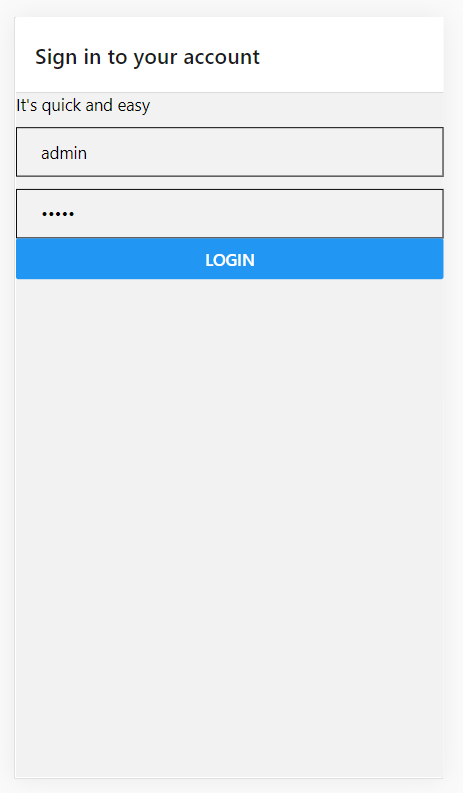
## **6.1.3 Result of Case 3 (IPad)**

User enters the amount 39991.33 earned from IInovative Creators and the final result after clicking ADD GIG button is:

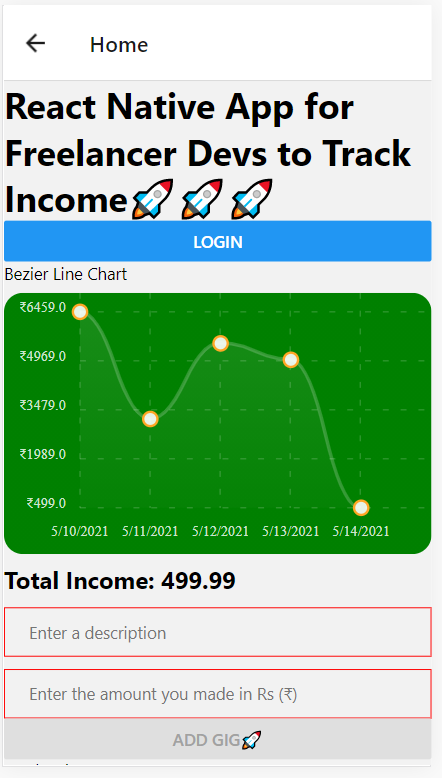
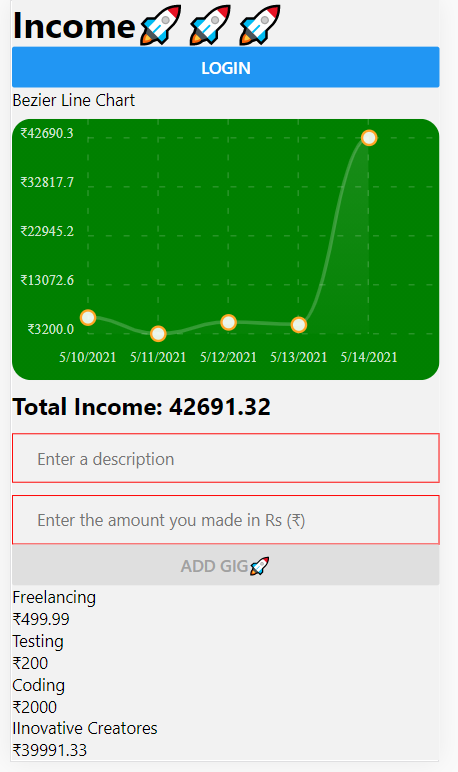


## **Android view of the application(Galaxy S5)**

LoginPage

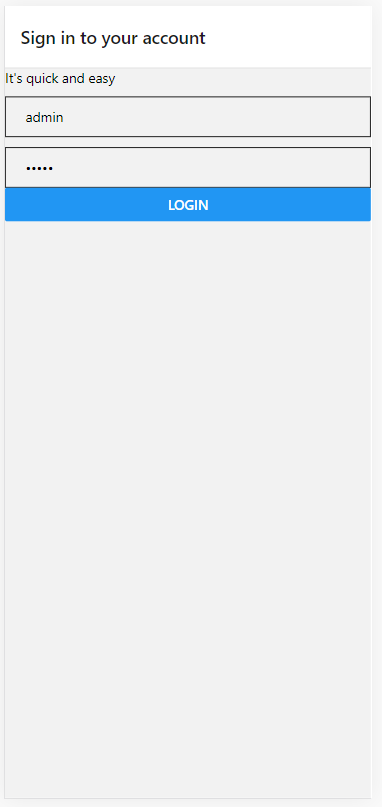
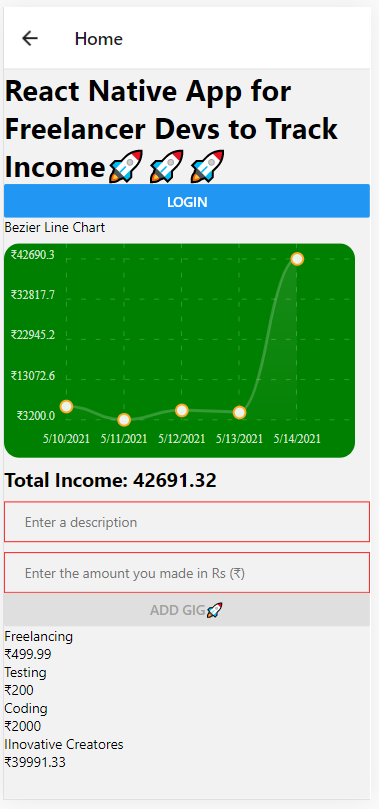


HomePage

## 6.3 **Iphone view of the application(iPhone X)**

Login page

# **Conclusion and Future work.**

Thus, we will be developing such a type of native application which helps users to reduce their effort of remembering their daily income. The app is capable of updating the income through any platform (i.e Android phone or Iphone or Desktop) also the application will have various features of automatically updating the users income on the graph and they can just know the total income they made on a particular day. In short the application will help its users to track their income daily.

Future work includes:

1. In further days, users can register and login to the native app.
2. There will be expense tracking done in the app.
3. There will be mails embedded with the app so that the users can open their app using their mail id as well.
4. Any change done in the app the user will get mail on the registered mail id.
5. Also, backup details will be recorded on the database.

# **References**

WEBLIOGRAPHY

|  |  |
| --- | --- |
| [1] | Hrithik Gupta, Anant Prakash Singh, Navneet Kumar, J. Angelin Blessy, “Expense Tracker: A Smart Approach to Track Everyday Expense,” EasyChair Preprint, 2020. |
| [2] | Vaishnavi kolhe, Brishti Basu, Vivek Shah, Ayush Ostwal, “My Expenses,” International Research Journal of Engineering and Technology (IRJET), 2020. |
| [3] | N.ZahiraJahan MCA., M.Phil., K.I.Vinodhini, “Personalized Expense Managing Assistant Using Android”, International Journal of Computer Techniques, 2016.  BILIOGRAPHY  https://www.slideshare.net/RashnaMaharjan2/daily-expense-tracker-153160282  https://www.wrike.com/project-management-guide/faq/why-should-i-use-budget-tracking-in-project-management-software/  https://easychair.org/publications/preprint\_open/73S7  https://www.irjet.net/archives/V7/i2/IRJET-V7I2224.pdf  <http://oaji.net/articles/2017/1948-1513926576.pdf>  https://www.filemakr.com/btech-final-year-project-report-daily-expense-tracker |

# **Annexure – Published Paper**