

KNOWLEDGE & TECHNOLOGY
Bangladesh Army University of Engineering & Technology



Department of Computer Science and Engineering

A project report on
ProconnectEase

A project is submitted in partial fulfillment of the requirements of the degree of
Bachelor of Science in Computer Science and Engineering.

Submitted by

Name: Bibek Bhowmick
ID:20204010
Name: Najmun Nafiz
ID: 20204016
Name: Syeda Fairooz Nawal
ID:20204020

Supervised by

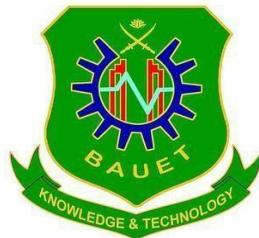
Mithun Kumar
Assistant Professor, Dept. of CSE, BAUET
Department of Computer Science and
Engineering
Bangladesh Army University of Engineering &
Technology
June, 2024

Submitted by
Mohammad
Golam Sarwar Bhuyan
Professor and Head,
Dept. of CSE, BAUET
Department of Computer Science and Engineering
Bangladesh Army University of Engineering &
Technology
June, 2024

KNOWLEDGE & TECHNOLOGY

Bangladesh Army University of Engineering & Technology

Department of Computer Science and Engineering



Approval

This is to certify that the project entitled "**ProConnectEase**" by has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering on April, 2024.

Signature of Supervisor(s)

.....

.....

.....

.....

Mohammad Golam Sarwar Bhuyan
(Professor & Head)
Department of CSE

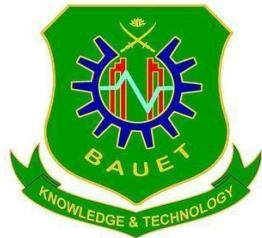
BAUET

Mithun Kumar
Assistant Professor
Department of CSE
BAUET

KNOWLEDGE & TECHNOLOGY

Bangladesh Army University of Engineering & Technology

Department of Computer Science and Engineering



DECLARATION

We hereby declare that project titled “**ProConnectEase**” is a great Project. We also ensure that it does not previously submit or published elsewhere for the award of any degree or diploma.

The work has been accepted for the degree of Bachelor of Science in Computer Science and Engineering at Bangladesh Army University of Engineering & Technology (BAUET).

Signature of Author (s)

.....

Bibek Bhowmick

Najmun Nafiz

Syeda Fairooz Nawal

ACKNOWLEDGMENT

First and foremost, we would like to thank to our supervisors, **Mohammad Golam Sarwar Bhuyan**, Professor and Head & **Mithun Kumar**, Assistant Professor , Department of Computer Science and Engineering, Bangladesh Army University of Engineering & Technology, Qadirabad who has been a source of tireless and patient guidance, advisement, and knowledge. This project could not be completed without his best directions, advice, and novel ideas.

We are very grateful to our respective teachers of Department of Computer Science and Engineering, Bangladesh Army University of Engineering & Technology, Qadirabad for their valuable suggestions, extending facilitation and inspiration time to time.

We are cordially thankful to laboratory staff of CSE department for their co-operation and amiable behavior.

Finally, we wish to extend our appreciation and thank to our family specially our beloved parents and all our well-wishers for their unconditional love, prayer and support.

ABSTRACT

The ProConnectEase is a ReactNative-based application platform designed to connect job seekers with relevant employment opportunities and assist them in their career development journey. It serves as a central hub for job seekers to explore a wide range of job listings, access career resources, and receive personalized job recommendations. Key features include user profile creation, job matching algorithms, and user engagement tools like job alerts and resume builders. It aggregates job listings from various sources, ensuring a comprehensive database of employment opportunities across industries and locations. It utilizes advanced matching algorithms to analyze user profiles and job listings, providing tailored job recommendations that align with each user's skills, qualifications, and preferences. In addition to job listings, the ProConnectEase also offers a wealth of career-related content, including expert advice, resume templates, interview tips, and industry insights. Users can engage with the platform through interactive features, such as job application tracking and community forums for networking and knowledge sharing. Data privacy and security are paramount, with robust measures in place to protect user information and ensure a safe online environment. ProConnectEase is committed to inclusivity and accessibility, offering support for users with disabilities and providing equal opportunities for all. Overall, the Job Assistance Portal aims to empower job seekers in their quest for meaningful employment, facilitate their professional growth, and bridge the gap between employers and talent. It plays a pivotal role in shaping successful career and fostering a thriving job market.

List of Contents

Chapter Title	Page No
Approval	i
Declaration	ii
Acknowledgment	iii
Abstract	iv
List of Content	v
List of figures	vi
List of Tables	vii

Contents

PO and KPs	1
Weighted of PO With Respect to Project Aims.....	2
Chapter 1	3
INTRODUCTION.....	3
1.1 Introduction.....	3
1.2 Motivation.....	3
1.3 Objectives	4
1.4 Expected Outcomes	4
1.5 Project Management And Finance.....	6
Chapter 2	29
SYSTEM ANALYSIS.....	29
2.1 Preliminaries/Terminologies	29
2.2 Related Works.....	30
2.3 Comparative Analysis	31
2.4 Scope of the Problem	32
2.5 Cjobenges.....	33
Chapter 3	34
Requirement Specification	34
3.1 Business Process Modeling	34
3.2 Requirement Collection and Analysis	37
3.3 Logical Data Model.....	39
Use Case Diagram	39
3.4 Design Requirement	40

Chapter 4	46
 Design Specification	46
4.1 Front-end Design and Backend Functionality	46
4.2 implementation Requirements	57
Chapter 5	58
 Code 58	
5.1 App.js:	58
5.2 Index.js:	61
5.3 AddPostScreen.js:.....	62
5.4 ApplyForum.js:.....	71
5.5 ChatBot.js:.....	80
5.6 HomeScreen.js:.....	82
5.7 ImageInput.js:.....	100
5.8 Input.js:.....	103
5.9 Job_des.js:	104
5.11 Job_explore.js:.....	125
5.12Job_record.js:.....	182
5.13JobPost.js:.....	213
5.14Login.js:.....	255
5.15 Profile.js:	270
5.16 Top.js:.....	272
5.17 Middle.js:.....	274
5.18 ResumeForm.js:.....	279
5.19 ShowCv.js:	292
5.20 SignUp.js:	298
5.21 Test result and report:	470
Chapter 6	474

Impact on Society, Environment and Sustainability	474
6.1 Impact on Society	474
6.2 Impact on Environment	475
6.3 Ethical Aspects	476
6.4 Sustainability Plan.....	477
 Chapter 7	 478
 At a Glance Whole project	 478
7.1 At a glance whole project:	478
 Chapter 8:	 480
 CONCLUSION & FUTURE SCOPE	 480
8.1 Discussion and Conclusion.....	480
8.2 Scope for Future Developments	480
 REFERENCES.....	 481

List of Figures

Figure No.	Title	Page No.
1.5.2.1	Feasibility study as part of the conception phase.	5
1.5.3.1	Program scheduling and Planning of July Month	6
1.5.3.2	Program scheduling and Planning of August Month	8
1.5.3.3	Program scheduling and Planning of September Month	9
1.5.3.4	Program scheduling and Planning of October to December Month	11
1.5.3.5	Program scheduling and Planning of January Month	10
1.5.3.6	Program scheduling and Planning of February Month	13
1.5.3.7	Program scheduling and Planning of March Month	14
1.5.3.8	Program scheduling and Planning of April Month	15
1.5.7	Cost Control	18
4.1.1	Business Process Modeling	33
4.1.2	Waterfall Model	34
3.3.1	Use-Case Diagram of ProConnectEase System	37
3.4.1	UX Welcome Screen	38
3.4.2	UX for Login	38
3.4.3	UX for SignUp	39
3.4.4	UX for Profile Page	39
3.4.5	UX for Job Post	40
3.4.6	UX for Job Search	40
3.4.7	UX for Job Description	41
3.4.8	UX for Video Resume Profile	41
3.4.9	UX for Video Resume Record	42

3.4.10	UX for Advanced Chatbot	42
4.1.1	Welcome Page	45
4.1.2	Home Page	45
4.1.3	User Login Page	46
4.1.4	User Signin Page	46
4.1.5	Job Searching	47
4.1.6	Job Description	47
4.1.7	ApplyForum	48
4.1.8	Apply to Company	48
4.1.9	Profile Page	49
4.2.0	Job Post	49
4.2.1	Add Post	50
4.2.2	Nearby Company Tracking	50
4.2.3	Resume Builder	51
4.2.4	Video CV	51
4.2.5	Advanced Custom Chatbot(Datasource:Bdjobs)	52
4.2.6	User Database structure	51
4.2.7	Post Database structure	53
4.2.8	Use case of customized chatbot	54

List of Figures

Table No.	Title	Page No.
5.2.1	Testing Result and report	468
5.2.2	Login test case	469
5.2.3	Member module test cases	470
5.2.4	Product module test case	471

PO and KPs

SN	Description Statement	PO	KP
1	Understanding the concept of web portals, drawbacks of the existing web portal and an attempt to integrate different features of famous web portals into one web portals after analyzing different systematic and theory-based software models.	PO2, PO4, PO10, PO12	K2, k3, k4, k8
2	Analyzing, understanding the design and continuously improving the User interface of the web portal so that it is more friendly and easy to use for any group of people	PO3, PO4 PO5, PO10, PO12	K5, K6, K7, K8
3	Understand the concept of video calling and how to add more people to initiate a conference call without the use of a third-party app or service	PO5, PO6, PO7, PO8, PO12	K2 K5 K6, K7 K8
4	Demonstrating and Analyzing ways for making a better environment for conducting live interviews and adding third party services like Google services, IDEs etc	PO3, PO5 PO6, PO7, PO8, PO12	K3, K5 K6, K7 K8
5	Ensuring secure handling of personal information by adapting to and keeping up with the advancement of technology and maintaining privacy of the users by ethically handling the security of the data.	PO5, PO6, PO7, PO8, PO12	K6, K7 K8
6	Analyzing the importance of Extra features like resume builders, interview tips and industry insights with and attempt to help users create perfect resumes, have great idea about the industry itself and have knowledge for attending interviews confidently	PO2, PO5(k6), PO10, PO11	K2, K5, K6, K7, K8
7	Giving presentation about the already build application which was build by as a team to demonstration and make users understand different features of the application	PO5, PO6, PO9, PO10 PO11	K5, K6, K7, K8
8	Interaction between employers and employee via audio call, video call, conference call along with posting contents like text, video or audio	PO2, PO4, PO9, PO10, PO11, PO12	K2, K5, K6, K7, K8

Weighted of PO With Respect to Project Aims

	1	2	3	4	5	6	7	8	Total
Project Aims	Understand existing web portals for integration	User interface enhancement	Implementation of Video call and conference call	Creation of interview environment	Data security	Resume Builder and insight of industry	Employer and employee interaction along with posting contents	Presentation about the project both individually and team wise	(%)
PO2	60%					20%		20%	100%
PO3		50%		50%					100%
PO4	60%	30%						10%	100%
PO5		30%	20%	10%	10%	10%	20%		100%
PO6			20%	10%	40%		30%		100%
PO7			10%	30%	60%				100%
PO8			20%		80%				100%
PO9				20%			50%	30%	100%
PO10	5%	5%				40%	30%	20%	100%
PO11				20%		40%	20%	20%	100%
PO12	25%	20%	20%		15%			20%	100%

Chapter 1

INTRODUCTION

1.1 Introduction

Introducing ProConnectEase, a cutting-edge mobile application developed using React Native technology. ProConnectEase serves as a sophisticated job assistance portal, tailored to meet the evolving needs of both job seekers and employers in today's competitive job market. With its intuitive interface and robust features, ProConnectEase offers a seamless platform for users to explore job opportunities, build professional profiles, and engage in meaningful interactions with potential employers. This introduction sets the stage for a comprehensive exploration of the motivations behind the development of ProConnectEase, as well as the objectives and functionalities that define its essence as a transformative tool for career advancement.

In an era marked by rapid technological advancements and dynamic shifts in the employment landscape, the traditional methods of job search and recruitment are undergoing significant transformations. With the increasing reliance on digital platforms for professional networking and career development, there arises a pressing need for innovative solutions that streamline the process of connecting job seekers with employers.

1.2 Motivation

The motivation behind the inception of ProConnectEase stems from a deep-seated understanding of the challenges faced by job seekers and employers alike in navigating the intricacies of the modern job market. Traditional job portals often fall short in providing personalized and efficient solutions, leading to frustration and inefficiencies for both parties involved. With ProConnectEase, our primary motivation is to bridge this gap by offering a platform that not only facilitates job search and recruitment but also fosters meaningful connections and professional development opportunities. By leveraging the power of React Native, we ensure that ProConnectEase is accessible across various mobile devices, empowering users to engage with the platform anytime, anywhere. Moreover, ProConnectEase is driven by a commitment to innovation and excellence, constantly evolving to incorporate new features and functionalities that enhance user experience and deliver tangible results. Through ProConnectEase, we aspire to empower individuals to take control of their career trajectories, connect with opportunities that align with their aspirations, and unlock their full potential in today's competitive job market.

1.3 Objectives

Enhance Job Seeker Experience: The primary objective of ProConnectEase is to enhance the job seeker experience by providing a user-friendly interface for browsing job listings, creating professional profiles, and applying for relevant positions. The platform aims to streamline the job search process, making it easier for users to discover new opportunities and showcase their skills and qualifications effectively.

Facilitate Employer Engagement: ProConnectEase seeks to facilitate meaningful engagement between job seekers and employers by offering tools for employers to post job listings, review candidate profiles, and communicate with potential hires. The platform aims to create a conducive environment for employers to connect with qualified candidates efficiently, ultimately leading to successful recruitment outcomes.

Promote Networking and Professional Development: Another key objective of ProConnectEase is to promote networking and professional development among its users. The platform will feature networking functionalities, such as connecting with industry professionals and joining relevant groups and communities, to foster collaboration and knowledge-sharing opportunities. Additionally, ProConnectEase will offer resources for skill development and career advancement to empower users in their professional journeys.

1.4 Expected Outcomes

Expected outcomes are directly derived from the unit's objectives (the actions, strategies, and activities we provide). The process of identifying expected outcomes helps us deconstruct the mission further into changes we expect to result from our objectives. Expected outcomes help us describe the results of our work from the perspective of those who benefit.

The expected outcomes of a ProConnectEase can vary depending on the specific features and functionalities it offers, as well as the goals and requirements of the organization implementing it. However, here are some common expected outcomes:

- i. **Improved Job Matching:** One of the primary expected outcomes of ProConnectEase is to facilitate improved job matching between employers and job seekers. By providing advanced search and filtering options, personalized job recommendations, and comprehensive candidate profiles, the platform aims to increase the likelihood of successful matches that align with both parties' requirements and preferences.
- ii. **Increased User Engagement:** ProConnectEase anticipates increased user engagement among both job seekers and employers. Through its intuitive interface, interactive features, and robust networking capabilities, the platform aims to foster active participation and meaningful interactions, leading to enhanced user satisfaction and loyalty over time.
- iii. **Enhanced Recruitment Efficiency:** ProConnectEase aims to enhance recruitment efficiency for employers by streamlining the hiring process and reducing time-to-hire. By offering tools for job posting, candidate screening, and communication, the platform enables employers to

identify and connect with qualified candidates more efficiently, ultimately leading to faster and more successful recruitment outcomes.

- iv. **Empowered Career Growth:** Another expected outcome of ProConnectEase is to empower individuals in their career growth and professional development endeavors. Through access to resources for skill enhancement, networking opportunities, and career advancement guidance, the platform seeks to equip users with the tools and knowledge needed to progress in their chosen fields and achieve their career goals.
- v. **Expanded Professional Networks:** ProConnectEase anticipates facilitating the expansion of users' professional networks by connecting them with like-minded individuals, industry experts, and potential collaborators. By offering networking features such as group discussions, events, and messaging, the platform enables users to forge valuable connections and leverage them for career advancement and growth.
- vi. **Positive Impact on the Job Market:** Ultimately, ProConnectEase aims to have a positive impact on the job market by fostering a more efficient, transparent, and inclusive hiring ecosystem. By facilitating meaningful connections between job seekers and employers, promoting fair and equal opportunities, and empowering individuals to take control of their professional journeys, the platform strives to contribute to the overall improvement of the job market landscape.
- vii. **Enhance User Privacy and Security:** ProConnectEase aims to prioritize user privacy and security by implementing robust data protection measures, including encryption protocols, secure authentication mechanisms, and adherence to relevant data privacy regulations. The platform seeks to build trust among users by safeguarding their personal information and ensuring a secure environment for interactions and transactions.
- viii. **Promote Diversity and Inclusion:** ProConnectEase is committed to promoting diversity and inclusion within the job market by providing equal opportunities for individuals from diverse backgrounds, including those from underrepresented groups. The platform will implement measures to prevent discrimination and bias in the recruitment process and actively support initiatives that foster diversity and inclusion in the workplace.

By incorporating these additional objectives into the project scope, ProConnectEase aims to create a comprehensive and impactful solution that addresses the diverse needs and challenges faced by job seekers and employers in today's dynamic job market landscape.

1.5 Project Management And Finance

In project management and finance there are some topics such as:

Initial investigation, Feasibility Study, Program scheduling and planning, Progress tracking, Budgeting, Cost benefit analysis, Cost control, Risk management, Team management, Finance management, Change management. In this chapter we also discuss about all of these.

1.5.1 Initial Investigation

For ProConnectEase project many users know a problem exists but do not know what it is or how to explain it. Before committing resources to a full-fledged study, the user undertakes a short internal investigation to clarify the problem and evaluate the merit of solutions. The investigation starts with fact-finding interviewing managers and users, gathering data, and reviewing existing documentation. A clear statement of the problem is formulated when we implemented ProConnectEase project, objectives are defined, and a list of alternative, potential solutions is compiled. The investigation focuses on the elements of the problem, including:

- i. The environment.
- ii. The needs, symptoms, problem definition, and objectives.
- iii. Preliminary alternative solutions and where possible the estimated costs benefits, strengths, and weaknesses of each steps.
- iv. Affected individuals and organizations.

Based on the internal investigation the customer must decide whether or not to proceed with the idea. We need further study. To approve the concept for further study, the customer must be convinced that:

- i. The need is real and funding is available to support it.
- ii. The idea has sufficient priority in relation to opportunities presented by other ideas.
- iii. The idea has particular value in terms of, for example, applying new technology, enhancing reputation, increasing market share, or raising profits.
- iv. The idea is consistent with the goals and resources of the organization.

1.5.2 Feasibility Study

Feasibility is the process of investigating a need, problem, and solutions in sufficient detail to determine if a solution is economically viable and worth developing. The initial investigation described previously is a form of feasibility study, but itself is usually rather cursory and, hence, insufficient to commit to a project. The feasibility process

involves a more protracted, rigorous study that includes consideration of alternative solutions (system concepts) and the benefits and economic viability of each.

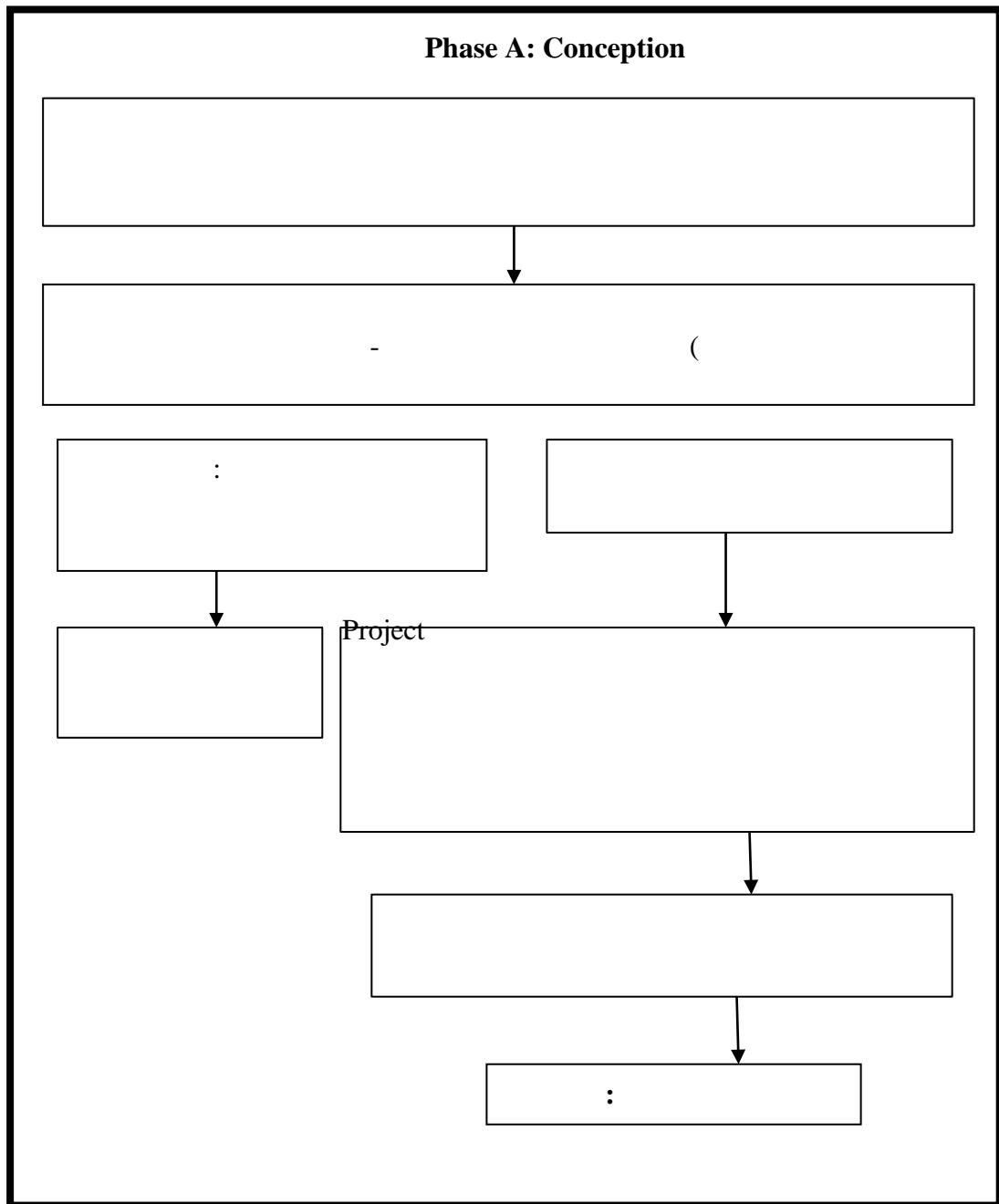
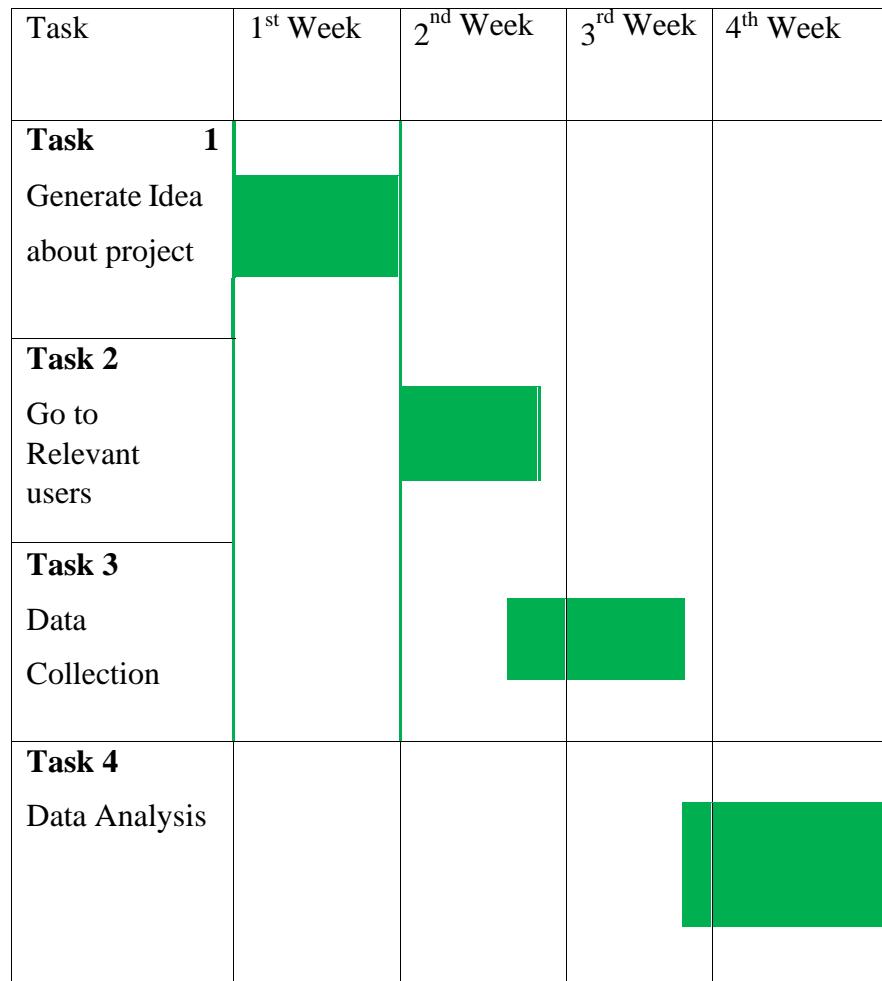


Figure 1.5.2.1: Feasibility study as part of the conception phase.

1.5.3 Program scheduling and planning

iii. Month: July

Task	1 st Week	2 nd Week	3 rd Week	4 th Week
Task 1 Generate Idea about project				
Task 2 Go to Relevant users				
Task 3 Data Collection				
Task 4 Data Analysis				



The Gantt chart illustrates the timeline for four tasks over a four-week period. Task 1 (Generate Idea about project) is completed in Week 1. Task 2 (Go to Relevant users) begins in Week 2. Task 3 (Data Collection) begins in Week 3. Task 4 (Data Analysis) begins in Week 4.

Figure: 1.5.3.1: Program scheduling and Planning of July Month

At first, our team members are called a meeting. In the meeting, we are decide to make ProConnectEase project . Then we are present our project idea to our course teacher. Our teacher accept our project, then we are planning to make our project. We are three persons in our team. They are Bibek Bhowmick, Najmun Nafiz and Syeda Fairooz Nawal.

July 1st Week: This week we are generate our idea. We are taking one week to generate our idea.

July 2nd week: This week we gather our requirement and different recommendation from different users starting from students of different departments like engineers, English students and BBA students. We also took advice from different Bankers, CEO of Byte capsule, Journalists and HRs of different Departments about what kind of features will be helpful for them to hire employees and what kind of features should be included for the employees to find jobs easily

July 3rd week, This week we collect data. In our project, we must need data. Our team member Najmnun Nafiz and Syeda Fairooz Nawal studied different job portals that are currently famous for finding jobs like Likedin, BDjobs, chakri.com, Xing etc to gather information about the drawbacks and potentials of different apps and how they work and how their combined features can be useful for building our application and making is even more user friendly.

July 4th week, This week we gathered our all data then we are analysis our data with our all team members.

iv. **Month: August**

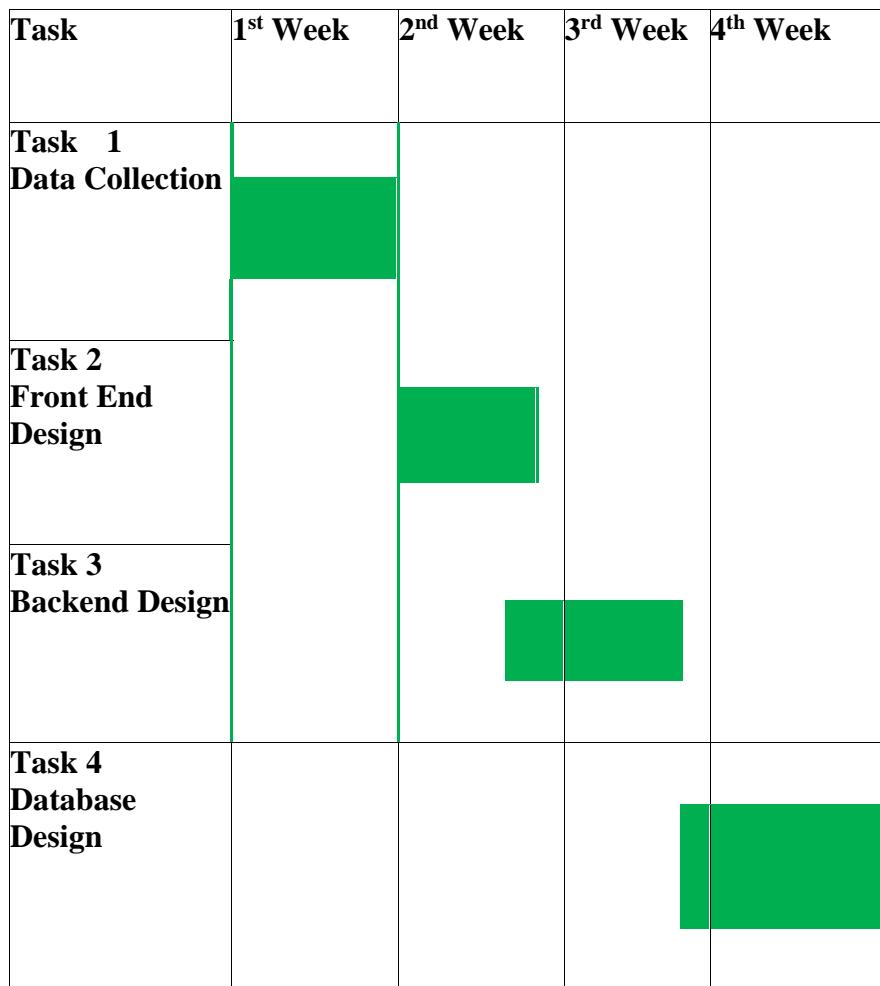


Figure: 1.5.3.2: Program scheduling and Planning of August Month

August 1st week, Last week after analysis our data we realize that we miss to gather some data so we investigated more to gather data online.

August 2nd week, This week we are done the front end design. We divided the entire application design into 19 pages and gave each member 6 to 7 pages to design.

August 3rd week, This week we are start to backed development. Bibek Bhowmick and Najmun Nafiz is doing the development site and structure. Some corrections of Ui/Ux design are given to our teacher this correction is solved Syeda Fairooz Nawal.

August 4th week, This week are doing database design. Bibek Bhowmick done the database model and create table. Najmun Nafiz doing the create table and input data. Syeda Fairooz Nawal done the connection between database and our project.

v. **Month: September**

Task	1 st Week	2 nd Week	3 rd Week	4 th Week
Task 1 Create Framework				
Task 2 Home page and navigation				
Task 3 Browser compatibility and responsiveness				
Task 4 Database Design				

```

graph LR
    subgraph Week1 [Week 1]
        T1[Task 1: Create Framework]
    end
    subgraph Week2 [Week 2]
        T1
        T2[Task 2: Home page and navigation]
    end
    subgraph Week3 [Week 3]
        T2
        T3[Task 3: Browser compatibility and responsiveness]
    end
    subgraph Week4 [Week 4]
        T3
        T4[Task 4: Database Design]
    end

```

Figure: 1.5.3.3: Program scheduling and Planning of September Month

September 1st week, In this week our team members create wireframes. Bibek Bhowmick and Najmun Nafiz doing this work.

September 2nd week, This week we are make home page and navigation. Syeda Fairooz Nawal done home page and Bibek Bhowmick complete home pages and navigations. Najmun Nafiz this week doing some work in wire frames.

September 3rd week, This week we are complete browser compatibility and do responsive. Syeda Fairooz Nawal done the responsive part. Bibek Bhowmick and Najmun Nafiz done the browser compatibility.

September 4th week, This week we are done some new requirements design and development. Bibek Bhowmick and Syeda Fairooz Nawal done design part. Najmun Nafiz and Bibek Bhowmick done the development part.

vi. **Month: October to December**

Task	1 st week	2nd week	3rd week	4th week	5th week	6 th week
Task 1 Define system objectives Modules and features						
Task 2 Some Database						

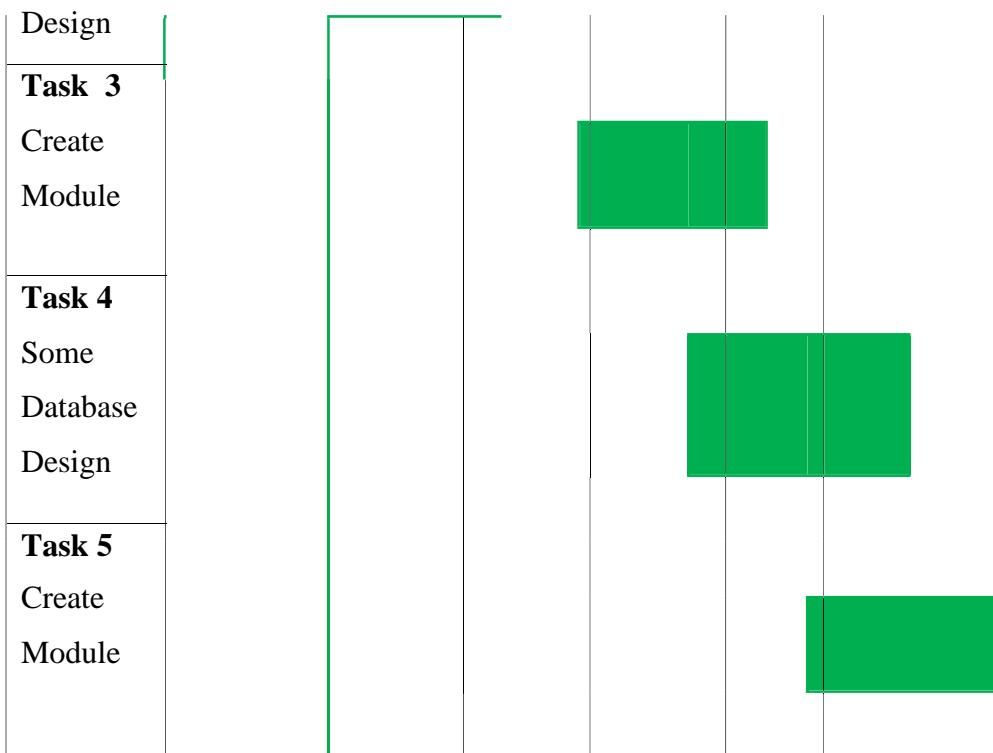


Figure: 1.5.3.4: Program scheduling and Planning of October to December Month

October 1st and 2nd week, This week we are done modules. Bibek Bhowmick and Najmun Nafiz done modules. Syeda Fairooz Nawal done previous task define system.

October 3rd and 4th week, This week we are complete some database design. . Bibek Bhowmick and Najmun Nafiz done database design. Syeda Fairooz Nawal done previous week task features.

November 1st and 2nd week, This week we are done module. Syeda Fairooz Nawal and Bibek Bhowmick are done module task. Najmun Nafiz done database design task.

November 3rd and 4th week to December 1st and 2nd week This month our team members are busy in the semester final exam. This week we complete just define system objective.

December 3rd and 4th week This week we are added more data set and design database. Bibek Bhowmick, Najmun Nafiz and Syeda Fairooz Nawal together done task module.

vii. **Month: January**

Task	1 st Week	2 nd Week	3 rd Week	4 th Week
Task 1				
Database				
Design				
Task 2				
Design User Profile				
Task 3				
Design Admin Profile				
Task 4				
Database Design				

Figure: 1.5.3.5: Program scheduling and Planning of January Month

January 1st week, This week we are done database design. This week Md. Tanvir Ahmed Tagim done database connection. Md. Moneruzjaman Mone and Fahmida Rahman Moumi done database design task.

January 2nd week, This week we are done user profile design task. Md. Moneruzjama Mone and Fahmida Rahman Moumi done user profile design. Md. Tanvir Ahmed Tagin done database task.

January 3rd week, This week we are done admin profile design. This week Fahmida Rahman Moumi and Md. Moneruzjaman Mone design admin profile. Md. Tanvir Ahmed Tagim set up admin and user profile.

January 4th week, This week we are done database design. This week Najmun Nafiz done database design. Bibek Bhowmick and Syeda Fairuz Nawal done profile design work.

viii. Month: February

Task	1 st Week	2nd Week	3 rd Week	4 th Week
Task 1 Gather some new information of job				
Task 2 Design a new home page				
Task 3 Design Home page				

Figure: 1.5.3.6: Program scheduling and Planning of February Month

February 1st week, This week we are gather more information. Our teacher given some new requirements in this causes we are gather more information.

February 2nd week, This week we are design a new home page. Syeda Fairuz Nawal done navigation bar. Najmun Nafiz done home page design.

February 3rd week, This week we are complete home page. Bibek Bhowmick and Najmun Nafiz done home page full design.

ix. Month: March

Task	1 st Week	2 nd Week	3 rd Week	4 th Week
Task 1 Change Some Designs				
Task 2 Create Dataset				
Task 3 Added new features				

The Gantt chart illustrates the timeline for three tasks over four weeks. Task 1, 'Change Some Designs', is completed by the end of week 2. Task 2, 'Create Dataset', begins in week 2 and is completed by the end of week 3. Task 3, 'Added new features', begins in week 3 and is completed by the end of week 4.

Figure: 1.5.3.7: Program scheduling and Planning of March Month

In March 1st 2nd 3rd and 4th week we do design chnage of ui ux. which we made with figma. Create dataset also and added some new features for our projct.

x. **Month: April**

Task	1 st Week	2 nd Week	3rd Week	4 th Week
Task 1 Add New Database				
Task 2 Added Video recording				
Task 3 Added Real time Company data				

Figure: 1.5.3.8: Program scheduling and Planning of April Month

In April 1st 2nd 3rd and 4th week we do database work, Video recording and added Real time data about different companies.

xi. **Month: April**

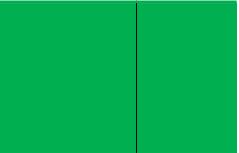
Task	1 st Week	2 nd Week	3 rd Week	4 th Week
Task 1 1 st Testing				
Task 2 Final Testing				
Task 3 Project Maintenance				
Task 4 Submit Project				

Figure: 1.5.3.9: Program scheduling and Planning of May Month

In May 1st 2nd 3rd and 4th week we do our project 1st testing, final testing, project maintenance and submit project.

1.5.4 Progress tracking

Tracking progress in the development of a ProConnect Ease involves monitoring various aspects of the project. Here's a general guideline for progress tracking:

Define Project scope:

The application will leverage the power of React Native to ensure cross-platform compatibility, enabling users to access its features from both iOS and Android devices. Advanced functionalities such as personalized job recommendations, resume building tools, and in-app messaging will be implemented to enhance user experience and increase engagement.

Create a project plan:

We break down the project into phases or sprints with a grant chart. Where we define tasks, allocate resources, and estimate timeframes for each task.

Use project Management tools:

Utilize project management tools like Android studio, Firebase to create tasks, assign responsibilities, and track progress.

Set Milestones:

We break down the project into phases or sprints with a grant chart and set milestones, vision control, continuous Integration

Regular time meeting:

Conduct regular team meetings to discuss progress, cjobenges, and upcoming tasks, code Reviews, testing, user acceptance testing, documentation, scalability and performance testing, feedback and improvement

1.5.5 Budgeting

1. Domain and hosting: Web Hosting and Domain Name: 11700 to 30,000 per year depending on the hosting requirements.

2. Security Measures: SSL Certificates: 1100 BDT to 20000 BDT per year.

3. Application Launching:

Google Play Store (Android):

a) Developer Account: Google Play Console Requires one-time registration fee of 2000 BDT. This fee grants you access to publish as many apps as you want to on the Google Play Store using developer account.

b) App Update: There is no additional fees for updating the Android app on the Play Store.

4. Maintenance and Update:

Ongoing Maintenance: 10,000 BDT to 20,000 BDT year for tools, plugins or services required for to maintain the functionality and security of the site.

Total Estimated Annual Ongoing Costs: BDT 24,000 to BDT 72,000

1.5.6 Cost Benefit Analysis

Total Estimated Annual Ongoing Costs: BDT 24,000 to BDT 72,000.

Total Revenue Calculation:

1.Premium Subscription Revenue:

Premium Subscription Revenue = Number of monthly subscriber * Monthly price

Let Number of Monthly subscribers 100 and the Monthly Price is 50 BDT
So, Subscription Revenue = $100 \times 50 \times 30 = 150,000$ BDT

2.Advertisement Revenue:

Advertisement Revenue = Monthly Active User (MAU) * Average monthly Ad Impressions per user * Click through Rate (CTR) * Revenue per thousand impressions (RPM)

Let monthly active users are 5000

Calculation of Average monthly Ad impression per user:

Ad impression means the number of times an ad appears for one user

Let for one user ads appear 50 times in 1 day,
then in one month it will be $50 \times 30 = 1500$

Calculation of CTR:

CTR = no clicks in ad/ page views

Let no of clicks in ad is 600 and the number of views in the page is 3000 Then CTR = $600/3000 = .2$

Calculation of RPM:

Revenue per 1000 Impressions (RPM) = (Estimated Earning / Number of Page views)

*1000 Let the cost per one click is 0.08 BDT, then

Estimated earning for one month = $600 \times 0.08 = 48$ BDT RPM = $48/3000 = 0.016$

So, Advertisement Revenue for one month = $5000 \times 1500 \times .2 \times .016 = 24000$ BDT So,
Advertisement Revenue for one year = $24000 \times 12 = 288,000$ BDT

Total earning from Revenue = Premium Subscription Revenue +Advertisement Revenue
 $= 150,000 \text{ BDT} + 288,000 \text{ BDT}$
 $= 438000 \text{ BDT}$

Total Benefit = $438000 - 72,000 = 366,000$ BDT

1.5.7 Cost Control

Cost control is an essential aspect of managing any project, including a ProConnectEase like job portal application project. Here are some steps and strategies you can consider to effectively control costs throughout the project lifecycle. Cost management is one of the most important responsibilities of a project manager; projects always need resources such as materials, labor and equipment, which generate costs. Those costs must be estimated and controlled throughout the project life cycle to complete the project.

1.5.7.1 Initial Budgeting: Start by creating a detailed budget that includes all the anticipated costs for the project. This should encompass hardware, software, personnel, third-party services, equipment, infrastructure, and any other relevant expenses.

1.5.7.2 Cost Estimation: Gather accurate estimates for each component of the project. Research and consult with experts to ensure your estimates are realistic.

1.5.7.3 Right Project Planning: Develop a detailed project plan that outlines the tasks, timelines, milestones, and associated costs. This will help you track progress and identify any deviations from the plan.

1.5.7.4 Quality Control: Ensure that the final product meets the required quality standards. Poor quality can lead to increased support and maintenance costs down the line.

1.5.7.5 Communication: Maintain open communication with stakeholders, including team members, clients, and sponsors. Transparency about costs and progress can help manage expectations.

1.5.7.6 Documentation: Keep thorough records of all expenses and financial transactions related to the project. This documentation is valuable for audits and future reference.

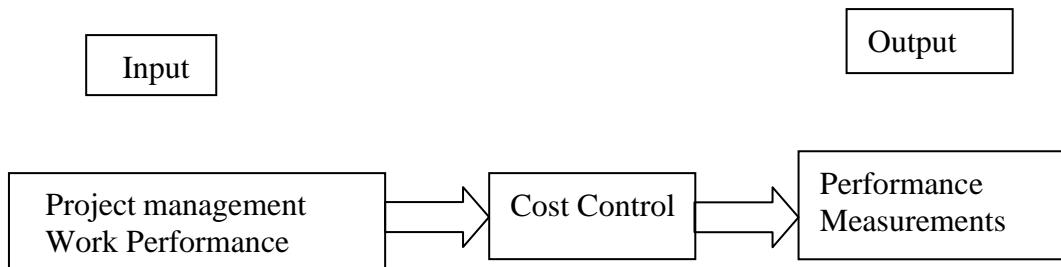


Figure 1.5.7: Cost Control

Cost control is the task of overseeing and managing project expenses and preparing for potential financial risks. This is typically the project manager's responsibility. Cost control involves managing the budget, as well as planning, and preparing for potential risks.

1.5.8 Risk Management

Implementing a ProConnectEase like job portal application can offer numerous benefits, such as improved efficiency, better security etc. However, like any technology project, there are also potential risks and challenges that need to be considered. Here are some of the risks associated with a ProConnectEase Job Portal App project:

- 1.5.8.1 Technological Risks: Technical challenges such as compatibility issues across different mobile platforms, software bugs, or unexpected system failures could delay the project timeline. Implement thorough testing procedures, including unit testing, integration testing, and user acceptance testing, to identify and address technical issues early in the development process. Regular code reviews and version control practices should also be implemented to ensure code quality and stability.
- 1.5.8.2 Market Competition: Intense competition from existing job assistance platforms or new market entrants could affect ProConnectEase's market penetration and user adoption rates. Conduct comprehensive market research to understand competitor offerings and identify opportunities for differentiation. Continuously monitor market trends and user feedback to adapt and enhance ProConnectEase's features and functionalities to remain competitive in the market.
- 1.5.8.3 Regulatory Compliance : Failure to comply with data protection regulations or industry standards could result in legal penalties or damage to ProConnectEase's reputation. Stay informed about relevant data privacy regulations, such as GDPR or CCPA, and ensure that ProConnectEase's data handling practices adhere to these regulations. Implement robust security measures, including encryption protocols and access controls, to protect user data from unauthorized access or breaches.
- 1.5.8.4 Resource Constraints: Limited availability of skilled personnel, technology infrastructure, or financial resources could impact ProConnectEase's development and implementation process. Conduct regular resource assessments to identify potential gaps or bottlenecks and develop contingency plans to address resource constraints. Consider outsourcing certain tasks or leveraging partnerships to access specialized expertise or additional resources as needed.
- 1.5.8.5 Initial Costs and Budget Overruns: Implementing a ProConnectEase system can involve significant upfront costs for hardware, software, and implementation. There is a risk of exceeding the budget due to unforeseen expenses or scope changes.

1.5.8.6 Maintenance and Upgrades: Technology evolves rapidly, and maintaining the system over time, including software updates replacements, can be cjobenging and costly.

1.5.8.7 User Experience Issues: If the user interface of the system is not intuitive or user-friendly, both employee and jobseekers may struggle to use the system effectively, leading to frustration and potentially impacting guest satisfaction.

1.5.8.8 Regulatory and Legal Compliance: Depending on the location and jurisdiction, there may be regulations and legal requirements related to data privacy, security, and technology usage that need to be carefully navigated.

We have to avoid all these risks. To effectively avoid and manage risks in a ProConnectEase project, you should adopt a comprehensive approach that encompasses various stages of the project lifecycle. Here are some strategies and steps to consider:

i. Thorough Planning and Risk Assessment:

Begin by identifying potential risks specific to your project. Engage employee, management, and technology experts, to list potential risks. Assess the impact and likelihood of each identified risk. This helps prioritize risks that need immediate attention.

ii. Data Security and Privacy:

Implement robust security measures to protect sensitive data. Comply with data privacy regulations and ensure that the system collects and stores data in a legally compliant manner.

iii. User Training and Adoption:

Provide comprehensive training for employee on how to use the smart system effectively. Address any concerns or resistance to change through proper communication and training programs. Design a user-friendly interface that minimizes the learning curve for both employee and everyone.

iv. Testing and Quality Assurance:

Conduct rigorous testing of the proconnectease system before full deployment. Identify and address any software bugs, technical glitches, or performance issues.

Perform testing in a controlled environment to simulate real-world scenarios and identify potential vulnerabilities.

v. **Regular Maintenance and Monitoring:**

Establish a schedule for regular system maintenance, updates, and security patches.

Monitor the system's performance, user feedback, and any emerging risks on an ongoing basis.

By following these steps and maintaining a proactive and collaborative approach, we can significantly reduce the risks associated with implementing a Proconnectease system and enhance the overall success of the project.

1.5.9 Team Management

Managing a Proconnectease requires effective team management to ensure smooth operations, technological advancements, and a positive experience for both employee and employer. Here are some key aspects to consider:

Cross Management system:

Development Team: Responsible for creating and maintaining the software and technological infrastructure.

Operation Team: Manages day-to-day activities, check-ins, check-outs, and ensures guest satisfaction.

Customer Team: Addresses user queries, technical issues, and provides assistance.

Marketing Team: Promotes the app, attracts users, and manages online presence.

Clear Role and Responsibilities:

- Clearly define roles and responsibilities for each team member.
- Ensure that everyone understands their role in contributing to the success of the Proconnectease.

Security and Privacy:

- Establish protocols to ensure the security and privacy of user information. Train the team on data protection and compliance with relevant regulations.

Flexibility and Adaptability:

ProconnectEase systems may evolve, so the team should be adaptable to changes. Encourage a mindset of continuous improvement and innovation.

1.5.10 Financial Management

Time Savings: A ProConnectEase can streamline the entire Job portal process, reducing the time required for manual tasks such as data entry, document verification, and communication with applicants. This time saved can be allocated to other important activities.

Estimated time savings: 100 hours per admission cycle.

Improved Efficiency: By automating various tasks, the system reduces the chances of errors and improves overall efficiency. It eliminates the need for paper-based documentation and manual record-keeping, leading to better organization and reduced administrative workload.

User Experience: A ProConnectEase provides applicants with a user friendly interface to submit their CV, track Companies.jobs and receive notifications. This improves the overall experience for applicants and reduces their frustration.

Cost Savings: While there are initial development and maintenance costs, over time, ProConnectEase can result in cost savings.

Time Savings: $100 \text{ hours} \times 60 \text{ TK} (\text{average hourly wage}) \times \text{number of Job seekers per year (assuming 300 admissions per year)} \times 3 \text{ years} = 540,000 \text{ TK}$

Cost Savings: 366,000 BDT

Based on this analysis, the net benefit of implementing The ProConnectEase of Bangladesh Army University of Engineering & Technology over a 1-year period is 366,000 BDTBDT. Therefore, the implementation of the system is financially advantageous, considering the estimated costs and benefits outlined above.

1.5.11 Change Management

Change management in project management is a combination of managing change and managing people (teams and stakeholders) to incorporate change. It significantly impacts how motivated employees are and how teams perform.

Identify the need for change: Clearly define why a change is necessary. Whether it's due to evolving requirements, technological advancements, or unforeseen circumstances, understanding the need for change is crucial.

Change Impact Assessment: Assess the potential impact of the change on various aspects of the project, such as scope, timeline, budget, and resources. Identify risks and opportunities associated with the change.

Change Implementation: Execute the change according to the strategy developed. This might involve updating project documents, modifying code, or integrating new processes. Ensure that everyone involved understands their roles and responsibilities during this phase.

1.5 Project Layout

A typical report layout for a Proconnectease system might include the following sections:

- i. Title Page: - The title of the report.
- ii. Table of Contents: A list of all the sections and subsections in the report along with their respective page numbers.
- iii. Executive Summary: A brief overview of the Proconnectease and its purpose. Summary of the key findings and recommendations. Any notable achievements or challenges faced during the reporting period.
- iv. Introduction: Background information about the Proconnectease. Objectives and goals of the system. Scope of the report and the reporting period covered.
- v. Requirement Specification: From this chapter we will know about the requirements of this project. Such as analysis, tools(which we are using in this project)
- vi. System Overview: Detailed description of the proconnectease system's architecture and components. Explanation of the functionalities and features provided by the system. Overview of the user roles and permissions of the system

Chapter 2

SYSTEM ANALYSIS

2.1 Preliminaries/Terminologies

In this section of the system analysis for ProConnectEase, we define and clarify the key terminologies and concepts relevant to the project. This includes defining terms such as user profile, job listing, messaging system, recommendation engine, and networking features. Clear definitions and explanations of these terms ensure a common understanding among project stakeholders and facilitate effective communication throughout the development process.

The preliminary background work for the ProConnectEase System involved:

- i. Requirement Gathering: The project team engaged in extensive discussions and consultations with the stakeholders to gather the requirements for the system. This included understanding the desired features, functionality, and user roles involved in the process.
- ii. System Design: Based on the gathered requirements, a comprehensive system design was created. This involved defining the architecture, database structure, user interface design, and integration points with other systems.
- iii. Technology Selection: The selection of appropriate technologies and tools was crucial for the successful implementation of the system. Factors such as scalability, security, and compatibility with existing infrastructure were taken into consideration during the technology selection process.
- iv. Development and Testing: The system development involved coding, database design, and user interface implementation. Regular testing and quality assurance were conducted to ensure the system's functionality, performance, and reliability.
- v. Security and Privacy Considerations: The protection of applicant data and maintaining the privacy of sensitive information were key considerations during the development process. Robust security measures, such as encryption, user authentication, and access control, were implemented to ensure the system's security.

The preliminary background work laid the foundation for the proconenectese System, setting the stage for its successful implementation and operation. This involved a thorough understanding of ProConnectEase system, collaboration with stakeholders, careful planning, and the utilization of appropriate technologies. The system's introduction aimed to enhance the efficiency, transparency, and accessibility of the process.

2.2 Related Works

"Related Works" typically refers to existing research papers, projects, or systems that are relevant to the topic under investigation. In the context of a ProconnectEase system, related works would include any previously published research papers, academic studies, case studies, industry reports, or existing systems that are directly or indirectly related to online job portal domains.

When conducting a literature review or exploring related works, you would typically look for:

- i. Research Papers: Academic papers or studies that have explored various aspects of ProConnectEase systems, such as resource allocation, scheduling algorithms, user experience design, optimization techniques, or other relevant topics.
- ii. Case Studies: Real-world examples or success stories of organizations or venues that have implemented ProConnectEase systems. These case studies can provide insights into cjobenges faced, solutions implemented, and the impact of the systems on operational efficiency.
- iii. Existing Systems or Solutions: Examining existing ProConnectEase systems, event software, or facility management platforms that offer features related to Chatting, Post, real time data, profile for user, reporting, and other relevant functionalities.
- iv. Industry Reports and Publications: Reports, whitepapers, or industry publications that discuss trends, best practices, or cjobenges in the field of linkedin, BDjobs, or related domains.
- v. Trade Journals and Magazines: Publications dedicated to the event management or facility management industry that feature articles, case studies, or expert opinions on job management systems.

By reviewing related works, researchers and practitioners can gain a deeper understanding of the existing knowledge, cjobenges, and solutions in the field of ProConnectEase systems. This knowledge can help inform the design, development, and implementation of new systems or improvements to existing ones.

2.3 Comparative Analysis

The Comparative Analysis section compares ProConnectEase with existing job assistance platforms and competitor offerings. This involves evaluating factors such as user experience, feature set, performance, and market positioning to assess ProConnectEase's strengths, weaknesses, opportunities, and threats relative to its competitors. By conducting a thorough comparative analysis, we can identify areas where ProConnectEase excels and areas where it may need to improve to gain a competitive edge in the market.

- i. User Experience: We assess the user interface (UI) and user experience (UX) design of ProConnectEase compared to competitors. This includes factors such as ease of navigation, intuitiveness of features, and overall aesthetics.
- ii. Feature Set: We compare the feature set of ProConnectEase with those of leading job assistance platforms such as LinkedIn, BDjobs, and others. This includes functionalities such as job search, profile creation, messaging systems, networking features, and additional tools like resume builders or skill assessments.
- iii. Performance: We evaluate the performance metrics of ProConnectEase, including loading times, responsiveness, and stability, in comparison to competitors. This ensures that the platform delivers a seamless and efficient user experience.
- iv. Market Positioning: We analyze the market positioning of ProConnectEase relative to competitors in terms of target audience, geographic reach, industry focus, and unique value proposition. This helps identify opportunities for differentiation and competitive advantages.
- v. Data Security and Integrity: In the manual process, there is a risk of data loss, unauthorized access, or tampering of documents. The online system ensures data security and integrity by implementing measures such as encryption, access controls, and secure storage of applicant information. This protects sensitive data and maintains the privacy of applicants.

By conducting a comprehensive comparative analysis across these key aspects, we can gain insights into ProConnectEase's strengths, weaknesses, opportunities, and threats relative to competitors. This analysis informs strategic decision-making and helps identify areas for improvement and differentiation to enhance ProConnectEase's competitiveness and value proposition in the job assistance market.

2.4 Scope of the Problem

The scope of the problem for the ProConnectEase System encompasses several key areas that need to be addressed in order to improve the efficiency and effectiveness of the jobmanagement process. These areas include:

- i. Manual and Paper-Based Processes: The traditional process at involves a significant amount of manual work and paperwork. This includes distributing and collecting physical application forms, manually reviewing and verifying documents, and maintaining paper-based records. The scope of the problem lies in the need to streamline these processes and reduce reliance on manual and paper-based methods.
- ii. Communication Gap: There is often a communication gap between applicants and the employer during the process. Applicants may face cjobenges in accessing timely updates. The scope of the problem is to bridge this communication gap and ensure smooth and transparent communication.
- iii. Accessibility and Inclusivity: The current process may present barriers for applicants from remote areas or with limited access to resources. Physical submission of application forms and documents can be cjobenging for such applicants. The scope of the problem is to make the process more accessible and inclusive by providing online platforms for application submission, document upload, and tracking, ensuring equal opportunities for all applicants.

By defining the scope of the problem in this manner, ProConnectEase aims to establish clear goals and objectives, allocate resources effectively, and ensure alignment with stakeholder expectations. This comprehensive scope sets the stage for the developmentof a robust and impactful job assistance platform that addresses the diverse needs of its target audience in the ever-evolving job market landscape.

2.5 Cjobenges

The Cjobenges section of the system analysis highlights potential obstacles and cjobenges that may arise during the development and implementation of ProConnectEase. This could include technical cjobenges such as scalability, compatibility, or performance optimization, as well as non-technical cjobenges such as user adoption, regulatory compliance, or market competition. By anticipating and addressing these cjobenges proactively, we can develop strategies and contingency plans to mitigate their impact and ensure the successful execution of the project.

- i. Technical Complexity: Developing a robust and scalable mobile application like ProConnectEase involves addressing various technical cjobenges, including compatibility across different mobile platforms (iOS and Android), optimizing performance for diverse devices and screen sizes, and implementing secure data storage and transmission protocols.
- ii. Data Integration and Migration: Integrating the ProConnectEasesystem with existing databases and systems within the institution can be complex. It requires seamless data transfer and synchronization to avoid discrepancies and ensure the accuracy of applicant information. Additionally, migrating data from the manual records to the digital system may require careful planning and execution to prevent data loss or corruption.
- iii. User Support and Helpdesk: As the ProConnectEase system is used by a large number of applicants, providing effective user support and helpdesk services becomes essential. Addressing queries, technical issues, and troubleshooting problems in a timely manner is crucial for a positive user experience. A well-organized helpdesk system and dedicated support team are necessary to handle user inquiries and resolve issues promptly.
- iv. Accessibility and Inclusivity: The ProConnectEase system should be accessible to all applicants, including those with disabilities or limited internet access. Ensuring compatibility with assistive technologies, providing alternative access methods, and addressing usability concerns are essential to promote inclusivity and equal opportunities for all applicants.

Implementing the Proconnnectese System comes with its fair share of cjobenges. Overcoming these cjobenges requires careful planning, technical expertise, user training, and continuous support.

Chapter 3

Requirement Specification

3.1 Business Process Modeling

The ProConnectEase incorporates business process modeling to streamline and optimize the Job management process. Business process modeling is a technique that allows for the visual representation and documentation of the various steps, activities, and interactions involved in a specific process. This system utilizes business process modeling techniques to map out the sequential flow of activities, decision points, and data exchanges involved in the admission process. This modeling helps in identifying bottlenecks, inefficiencies, and areas for improvement. Business process modeling enables the ProConnectEase Job portal application to:

- i. Identify Process Improvements: Through modeling, potential areas of improvement can be identified, such as eliminating redundant steps, automating manual tasks, or optimizing resource allocation. This helps in streamlining the process, reducing turnaround time, and enhancing overall efficiency.
- ii. Standardize Processes: The modeling process allows for the identification and documentation of standardized procedures and best practices for each stage. This ensures consistency and uniformity.
- iii. Enhance Communication and Collaboration: Business process models act as a visual representation that facilitates effective communication and collaboration among different stakeholders involved in the job management process. It provides a common understanding of the process, roles, and responsibilities, enabling better coordination and cooperation.
- iv. Analyze Performance and Metrics: Business process modeling allows for the integration of performance metrics and key performance indicators (KPIs) into the process. These insights can guide decision-making and continuous process improvement efforts.
- v. Support System Integration: Business process modeling helps in identifying the integration. This ensures seamless data exchange, integration of information, and interoperability between different systems involved in the admission process, such as student information systems or financial systems.
- vi. Facilitate Process Automation: By visualizing the process flow, business process modeling assists in identifying opportunities for automation. This includes automating routine tasks, generating automated notifications and alerts, and integrating decision support systems to streamline decision making processes.

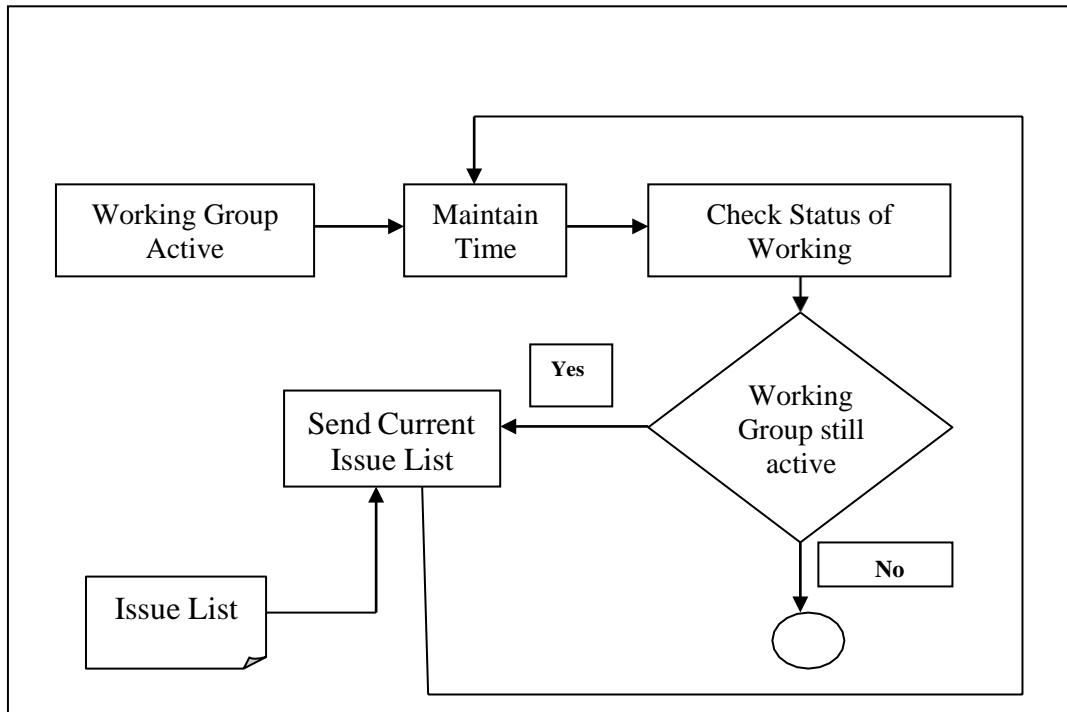


Figure 4.1.1: Business Process Modeling

It provides a comprehensive understanding of the process, facilitates process improvements, and enables data-driven decision-making. By continuously refining and optimizing the business processes through modeling, the system ensures a smoother and more streamlined management.

Waterfall model:

Waterfall approach is the first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. The following illustration is a representation of the different phases of the Waterfall Model. The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, and troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

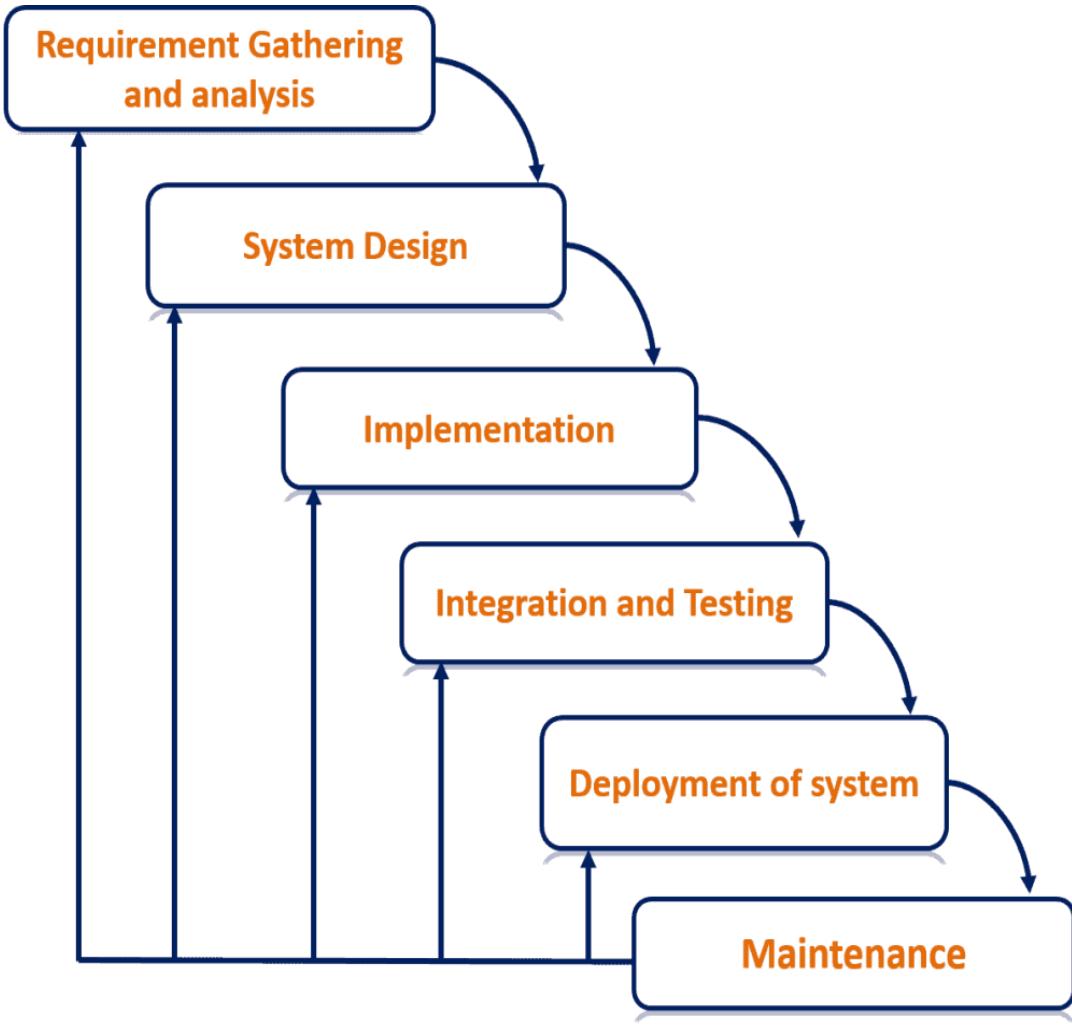


Fig 4.1.2: Waterfall Model

The waterfall model (figure) is characterized by its rigid and sequential nature, where each phase is completed before moving on to the next. This model works well for projects with well-defined requirements and where changes are minimal or unlikely. However, it may not be suitable for projects with evolving requirements or where early feedback and iterations are necessary.²⁶ While the waterfall model has its advantages, such as clear documentation and a structured approach, it also has limitations. The linear nature of the model makes it difficult to accommodate changes once a phase is completed, and any errors or misunderstandings in the early phases can have significant impacts on later stages.

3.2 Requirement Collection and Analysis

Requirements Gathering and Analysis is a crucial phase in the software development lifecycle, where the project team collaborates with stakeholders to identify, document, and analyze the requirements of the software system to be developed. This phase lays the foundation for the entire development process and plays a pivotal role in the success of the project.

- vii. Identifying Stakeholders: The first step in requirements gathering is to identify the stakeholders involved in the project. The process involves identifying everyone remotely connected with the project (individuals, groups, or organizations) that could impact or be impacted by a project's decision, activity, or outcome.
- viii. Gathering Requirements: Requirements gathering is the process of identifying your project's exact requirements from start to finish. This process occurs during the project initiation phase, but we'll continue to manage our project requirements throughout the project timeline. It involves gathering information about the desired functionalities, features, and constraints of the software system.
- ix. Documenting Requirements: Once the requirements are gathered, they need to be documented in a structured and organized manner. Requirement documentation typically includes functional and any other relevant information. Various tools and techniques, such as use cases, user stories, or requirement specification documents, can be used to document the requirements.
- x. Analyzing Requirements: Requirement analysis involves a careful examination and validation of the gathered requirements to ensure their completeness, consistency, and feasibility. This analysis helps identify any ambiguities, conflicts, or gaps in the requirements.
- xi. Managing Changes: This process includes documenting change requests, assessing their impacts on the project's scope, schedule, and budget, and obtaining stakeholder approval before incorporating the changes into the requirements.
- xii. Collaboration and Communication: Requirements gathering, and analysis is a collaborative effort that requires effective communication and collaboration among all stakeholders.
- xiii. Documentation Review and Sign-off: Once the requirements are analyzed and validated, they need to be reviewed by stakeholders for feedback and approval. This ensures that all stakeholders have a shared understanding of the requirements and agree before proceeding to the next phases of the software development lifecycle. Obtaining formal sign-off on the

requirements signifies the stakeholders' acceptance and commitment to the proposed solution.

Requirements Gathering and Analysis is a critical phase that sets the direction for the entire software development project. Successful requirements gathering and analysis contribute to the development of a software system that meets stakeholders' needs, aligns with business objectives, and forms the basis for subsequent development activities.

3.3 Logical Data Model

Use Case Diagram

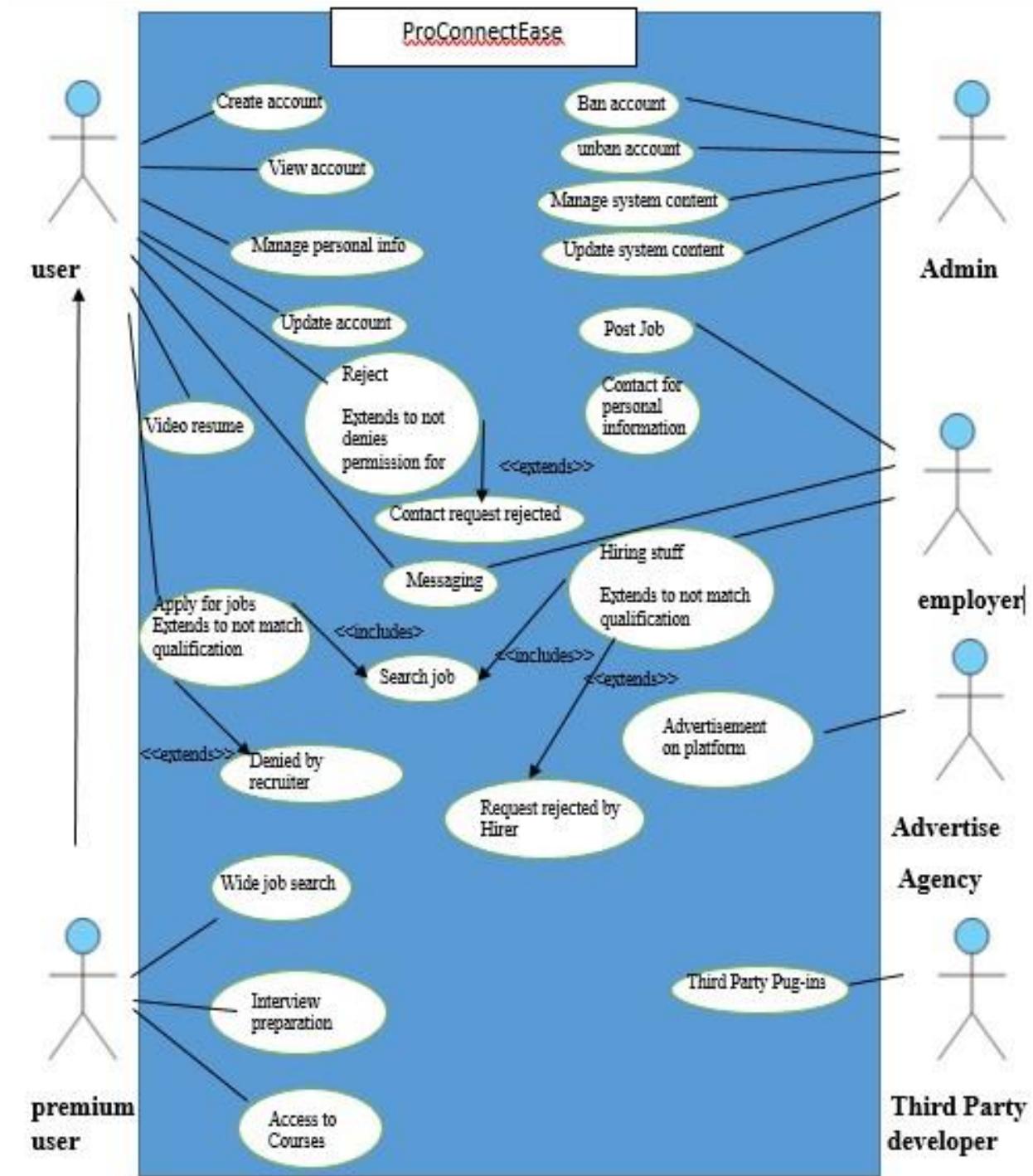


Fig 3.3.1: Use-Case Diagram of ProConnectEase System

3.4 Design Requirement

Where UX looks at the entire user experience and how everything ties together, interaction designers focus on user interactions and motion. Interaction designers apply UX fundamentals like design thinking, human-centered design, and user research (personas, case studies, etc.) to make decisions.



Figure 3.4.1: UX Welcome Screen for students

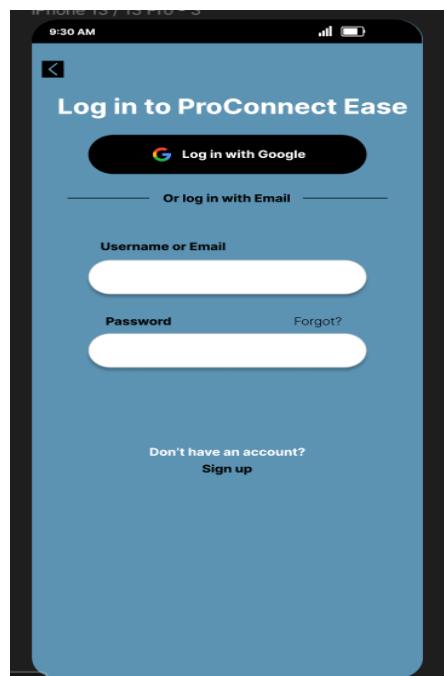


Figure 3.4.2: UX for Login

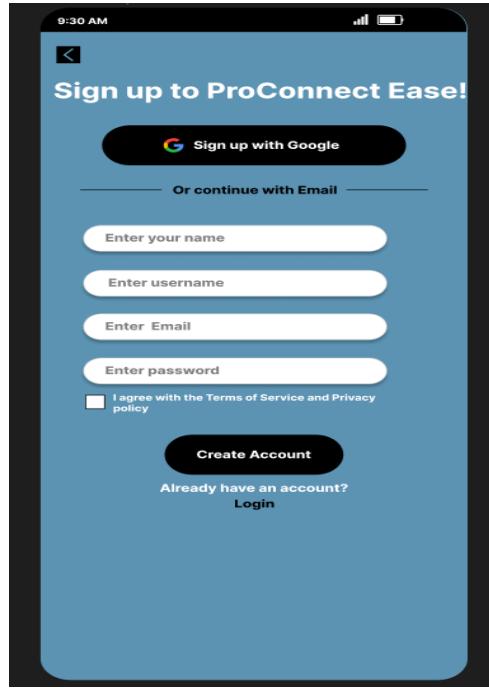


Figure 3.4.3: UX for SignUp

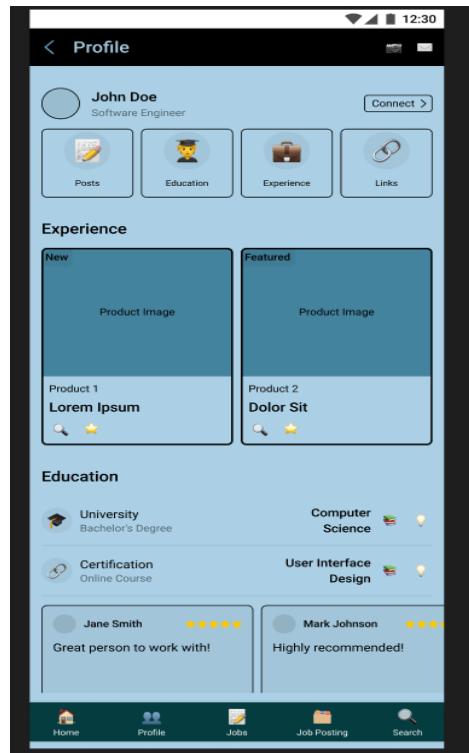


Figure 3.4.4: UX for Profile Page

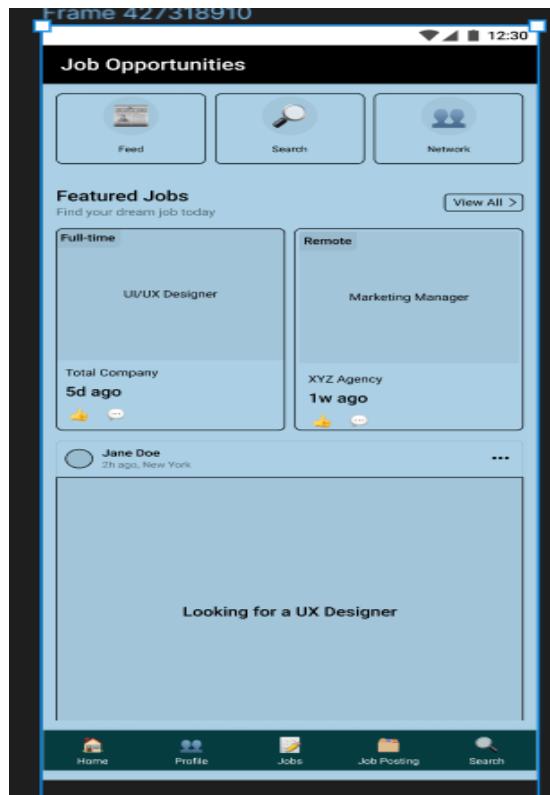


Figure 3.4.5: UX for Job Post

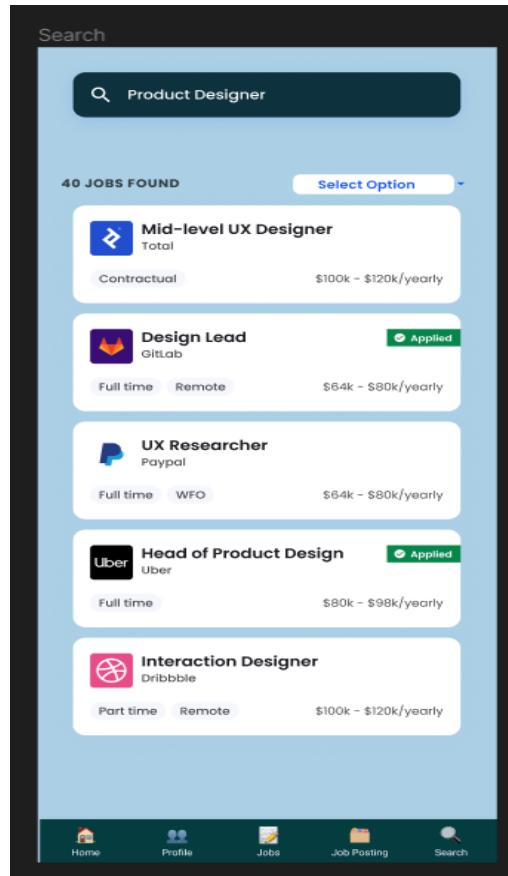


Figure 3.4.6: UX for Job Search

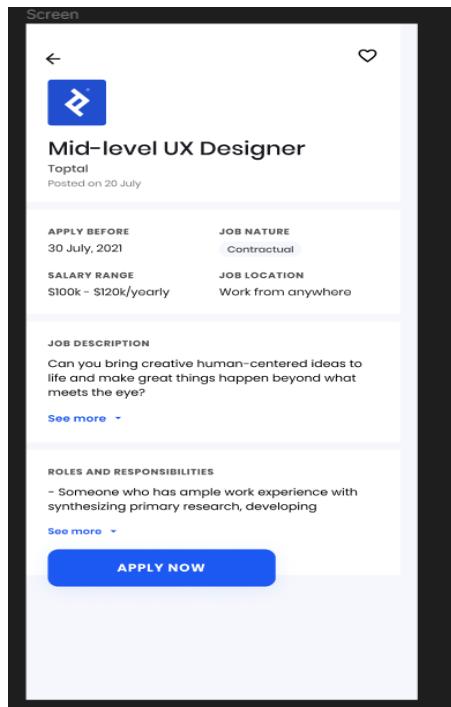


Figure 3.4.7: UX for Job Description

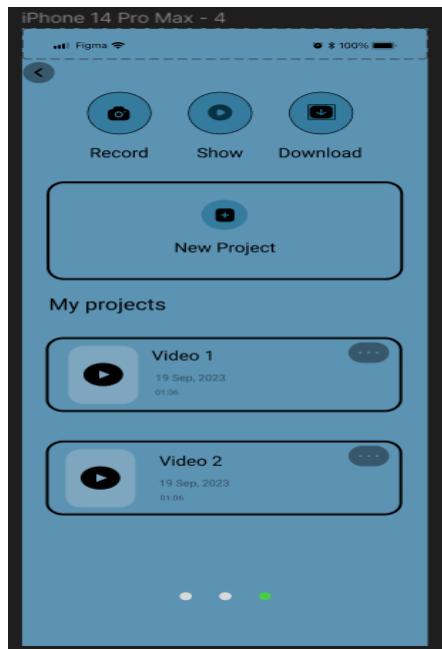


Figure 3.4.8: UX for Video Resume Profile



Figure 3.4.9: UX for Video Resume Record



Figure 3.4.10: UX for Advanced Chatbot

At the end of the business case stage of the design management in the project lifecycle, the owner should be better informed of all important design related issues and possible impacts on the project business case. Once the business case has been approved by the owner, the design manager can then proceed to the next stage of the project lifecycle, the outline design stage.

This stage involves clearly defining the owner design requirements so as to form a sound foundation for the design process to proceed and is the right time to engage consultants and set up the formal design management process. The following are the key tasks in this project lifecycle stage:

xiv. Define owner design requirements and project design needs

Further to the design related information gathered for the preparation of the business case, the design manager should gather any additional design related information from the owner and other relevant sources. Identify any gaps in the information provided and seek to source this information as appropriate. This is the right time to sit with the owner and review all the design related information and discuss the strategy forward to gather additional information.

xv. Engage design consultants

This is the time to now engage all the key design consultants that are required to develop the functional design brief and the concept design. It is critical that the consultant's scope of work is clear for the level of input required in their contract agreement.

xvi. Prepare the functional design brief and concept design

Manage and coordinate the design consultant team to deliver the functional design brief and the concept design that will respond to and record all the owner's requirements and needs and form the basis of the design to proceed for all the design disciplines. The functional design brief will be supported by concept design sketches and report, all of which provide an outline of the proposed design.

xvii. Prepare the design management plan

The design management plan provides the roadmap for the way the design will be managed and needs to be prepared at this stage of the design process for best results. It is a key component of the project management plan prepared by the project manager.

xviii. Prepare the outline design report

Finally and present the functional design brief, concept designs and the outline design process results to the owner as an outline design report. This

updates the owner, providing opportunity for comment prior to moving onto the next design phase.

Chapter 4 **Design Specification**

4.1 Front-end Design and Backend Functionality

Front-end UI, typically called front-end development or design, is the process of creating, testing and refining the user's side of a website or application. Some examples of front-end applications that enable web languages are HTML, JavaScript, and CSS. These three examples, as mentioned above, are essential to the development and accessibility of an interface. To implement front end design, follow these steps:

- i. Implement React Native Concepts(with html,css,js)
- ii. Sketch wireframes or create mockups
- iii. Design the user interface
- iv. Create HTML structure
- v. Apply CSS style
- vi. Implement interactivity with JavaScript
- vii. Integration with Backend
- viii. Optimize for responsiveness
- ix. Test and iterate
- x. Integrate with back-end
- xi. Expo Go environment
Compatibility
- xii. Continuously improve and iterate



Figure 4.1.1: Welcome Page

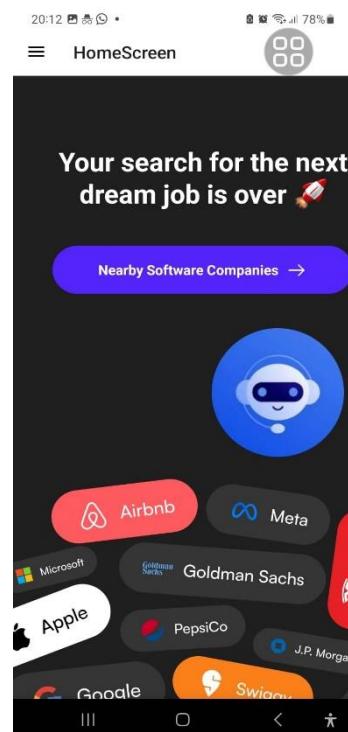


Figure 4.1.2: Home Page

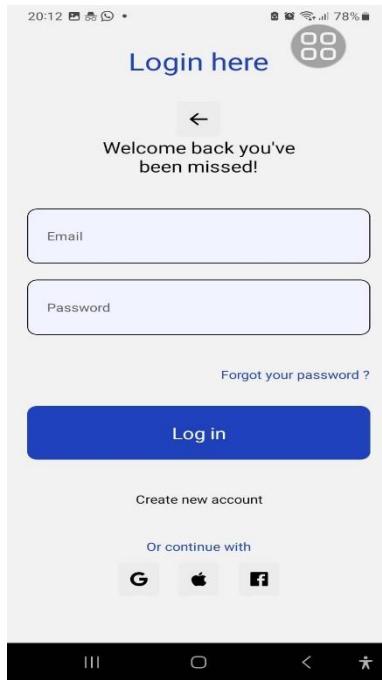


Figure 4.1.3: User Login Page

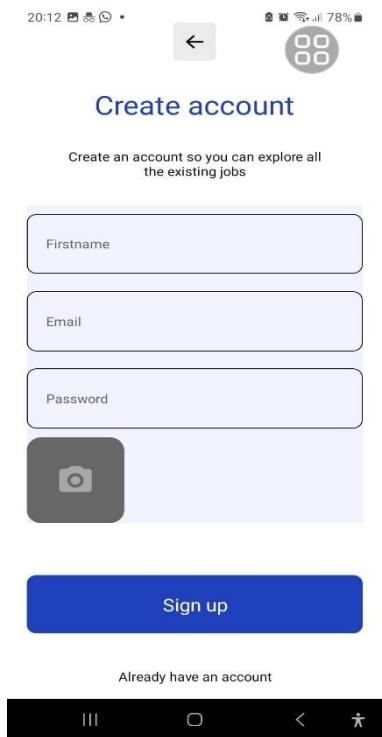


Figure 4.1.4: User SignIn

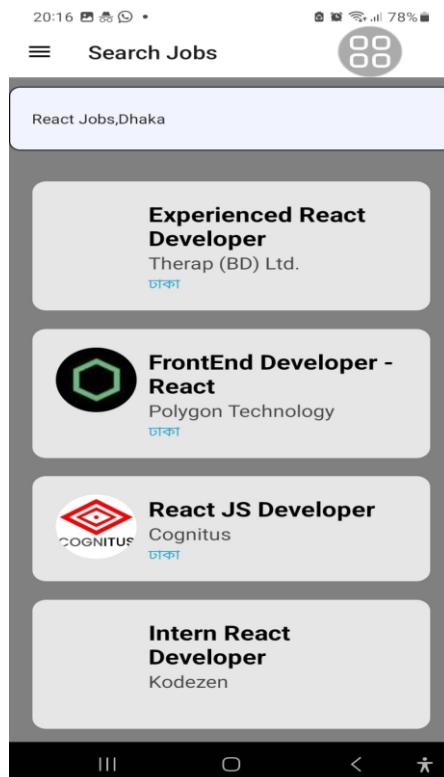


Figure 4.1.5: Job Searching

A screenshot of a job description page for a "Senior Mobile Developer (React Native/TypeScript)" position. The top navigation bar shows the time as 20:24 and battery level at 77%. The header includes a three-line menu icon, the job title "Job_des", a back arrow, and a large blue "APPLY NOW" button. Below the header is the company logo, which is a colorful stylized 'V'.

Senior Mobile Developer (React Native/TypeScript)
Virtasant
Posted on 2024-04-10T17:57:33.000Z
Job Quality Score: 0.6434
Job Apply Link: <https://bd.linkedin.com/jobs/view/senior-mobile-developer-react-native-typescript-at-virtasant-3889099483>

APPLY BEFORE 2024-05-10T17:57:33.000Z **JOB NATURE** Contract
JOB CITY **JOB TYPE** FULLTIME

JOB DESCRIPTION
Virtasant is a leading cloud consulting services provider. We heavily focus on lift & shift, cloud-native development, cloud cost optimization, and migration services. As a consulting company, we often are faced with the challenge of creating an engineering team in a matter of a week or two. To do that, we have created a secondary support business that runs an engineering talent network and provides staffing services.

The bottom of the screen features standard Android navigation icons: three horizontal bars, a square, a back arrow, and a person icon.

4.1.6: Job Description

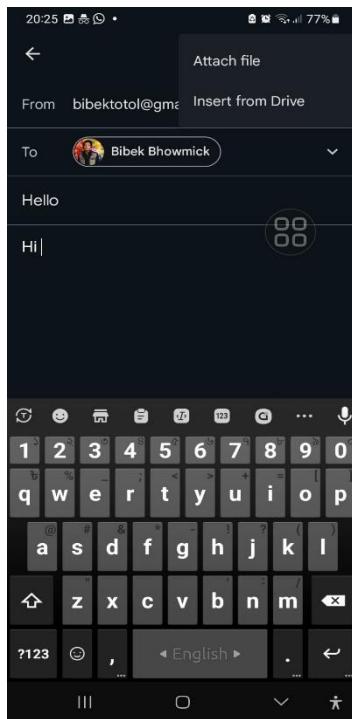


Figure 4.1.7: ApplyForum

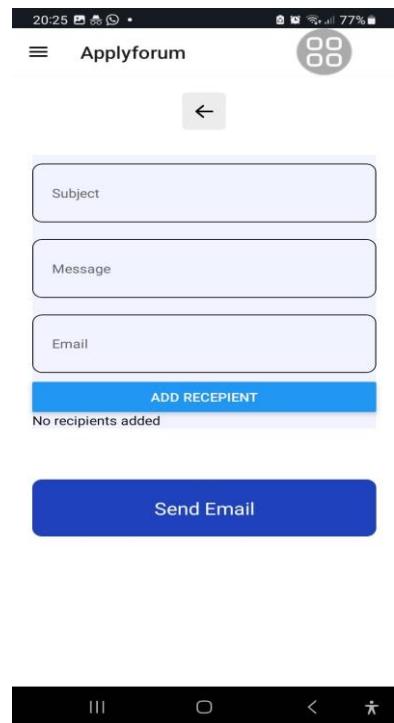


Figure 4.1.8: Apply to Company

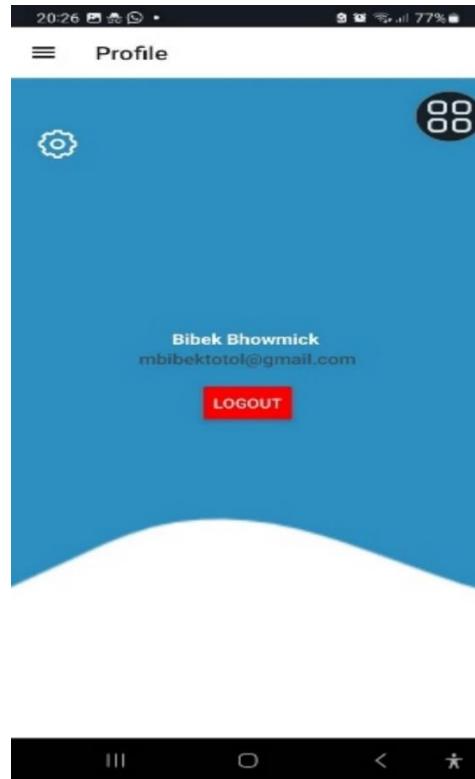


Figure 4.1.9: Profile Page

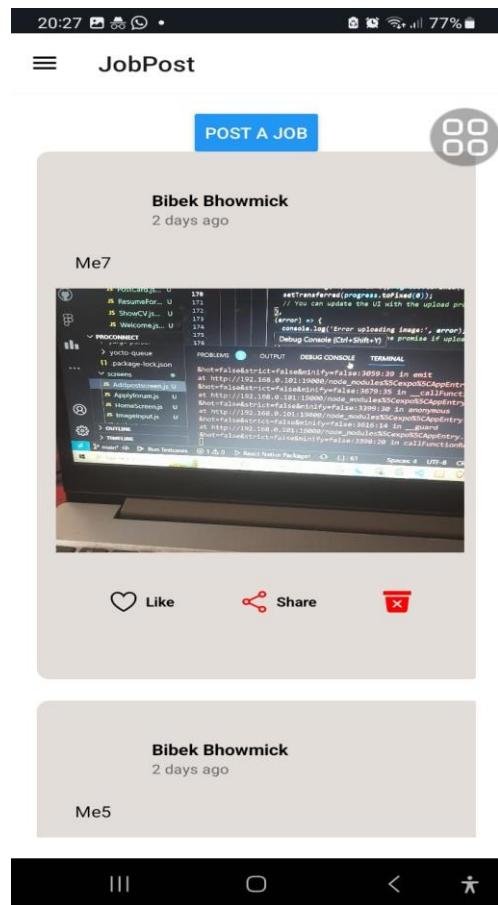


Figure 4.2.0: Job Post

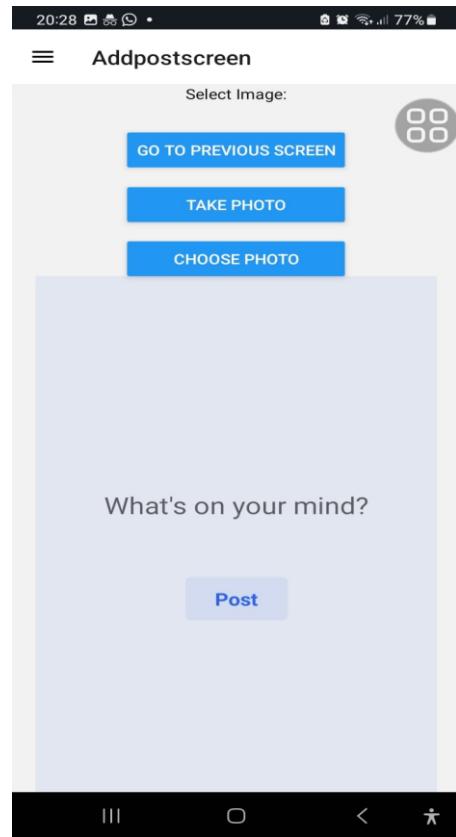


Figure 4.2.1: Add Post

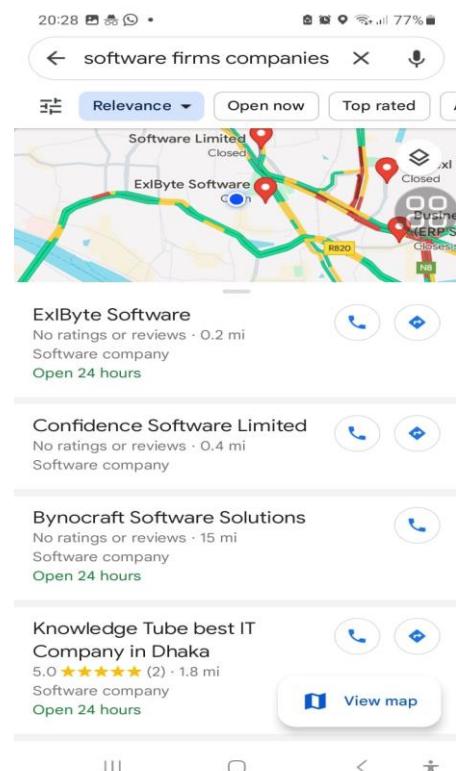


Figure 4.2.2: Nearby Company Tracking

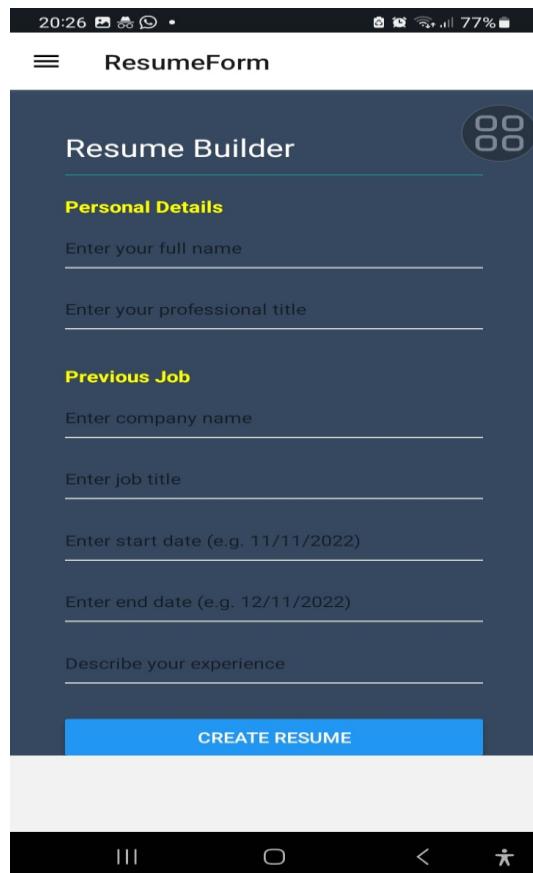


Figure 4.2.3: Resume Builder

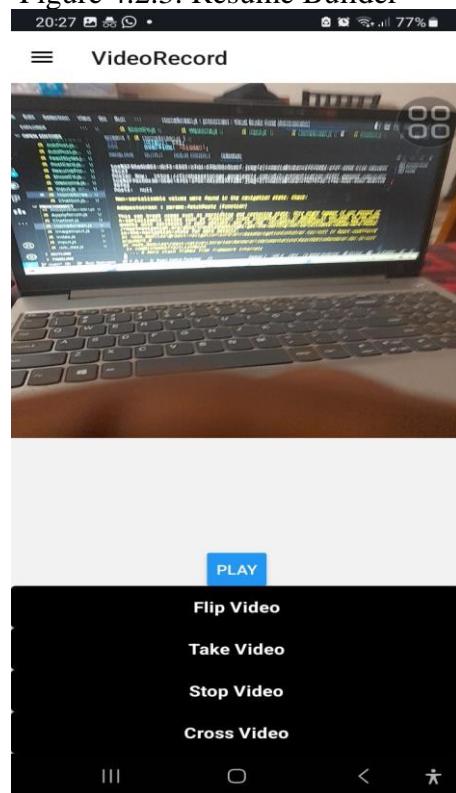


Figure 4.2.4: Video CV

Mamiya-OP (BD) Ltd.
Officer

Application Deadline: 30 Apr 2024

Apply Now ★ Shortlist Share: [f](#) [in](#) [g](#) Print

All Requirements Responsibilities Skills & Expertise Salary & Benefits Company Information

Summary

Vacancy: 1	Age: 27 to 35 years	Location: Chattogram
Salary: Negotiable	Experience: At least 3 years	Published: 7 Apr 2024

Applicants are encouraged to submit **Video CV**.

Requirements

Education

- Bachelor of Science (BSc) in Computer Science & Engineering

Experience

- At least 3 years
- The applicants should have experience in the following business area(s): Manufacturing (Light Engineering and Heavy Industry)

Bot
This chatbot was built surprisingly fast with Botpress
What do you want to know?

New messages

Mamiya-OP (BD) Ltd. which job they are currently offering and tell me the location,salary, age, education, and experience requirement

- Job Title: Senior Manager - Maintenance
- Location: Chattogram Export Processing Zone, Chattogram
- Salary: Negotiable
- Age: 35 to 45 years
- Education: B.Sc. in Industrial and Production Engineering (IPE) from any reputed university
- Experience Requirement: At least 10 to 12 years

Chat with bot

Figure 4.2.5: Advanced Custom Chatbot(Datasource: Bdjobs)

Database and UseCase Design:

The screenshot shows the Google Cloud Firestore interface. On the left is a sidebar with icons for authentication, Cloud Firestore, Cloud Functions, Cloud Storage, and Cloud Pub/Sub. The main area shows the 'users' collection under the 'authentication' project. A specific document, 'INdjV5dWL6c2A', is selected. The document details are as follows:

- email:** "nafiz@gmail.com"
- imageuri:** "file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540bib3d88-4662-b580-46bbf829f08d.jpeg"
- name:** "Nafiz"
- password:** "123456"

Figure 4.2.6: User database

The screenshot shows the Google Cloud Firestore interface. On the left is a sidebar with icons for authentication, Cloud Firestore, Cloud Functions, Cloud Storage, and Cloud Pub/Sub. The main area shows the 'posts' collection under the 'authentication' project. A specific document, 'ExHylF9IM8yrFg.', is selected. The document details are as follows:

- comments:** 0
- likes:** 0
- post:** "Me"
- postImg:** "https://firebasestorage.googleapis.com/v0/b/authentication-92f91.appspot.com/o/photos%2F7acb9c30-bc8c-4791-af94-441e9a747a84.jpeg?alt=media&token=f6df67aa-cf5e-4dac-9b6d-86891b876fa2"
- postTime:** 10 April 2024 at 05:07:55 UTC-7
- profileimg:** "file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540l3d04-4401-99f6-40c8156d7233.jpeg"
- userId:** "yhjs8tGpXMQSKbWoNp6ymxuUTep2"
- userName:** "Bibek Bhowmick"

Figure 4.2.7: Post database Structure

Beginning of the Conversation

This workflow is executed at the beginning of each conversation or when a conversation is not inside any workflow.

By default, this chatbot will answer user questions on the Start Node using the custom knowledge uploaded in the Knowledge Bases.

You can change this behavior in the **Knowledge Agent** settings.

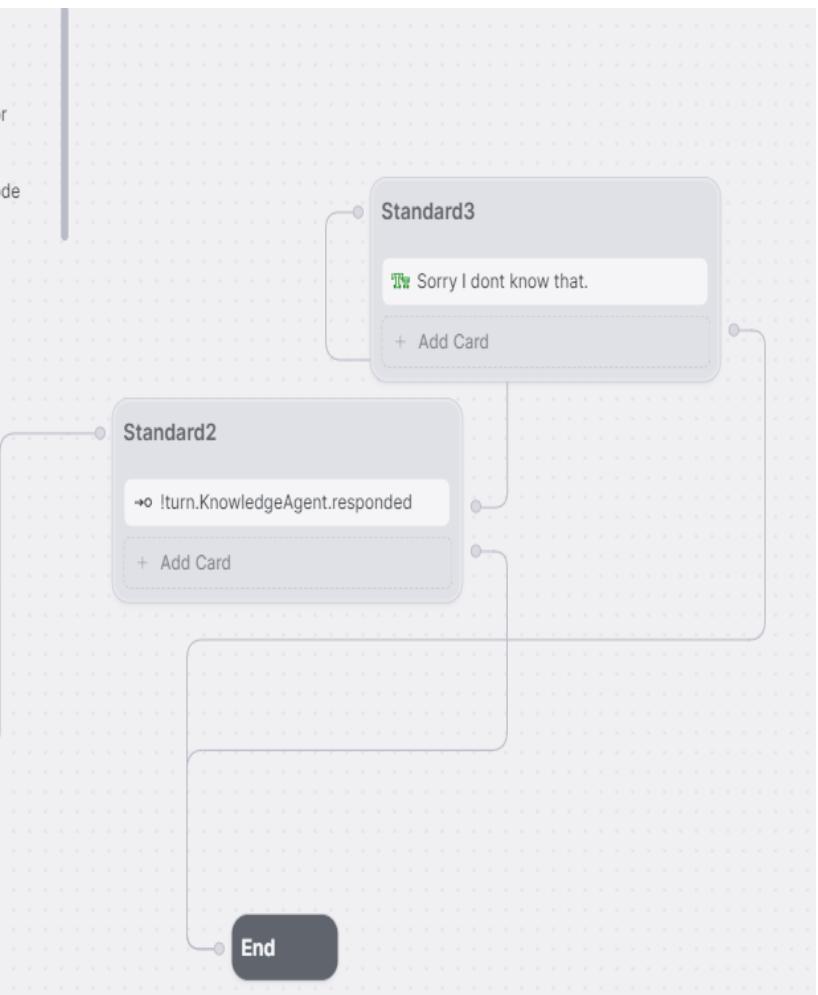
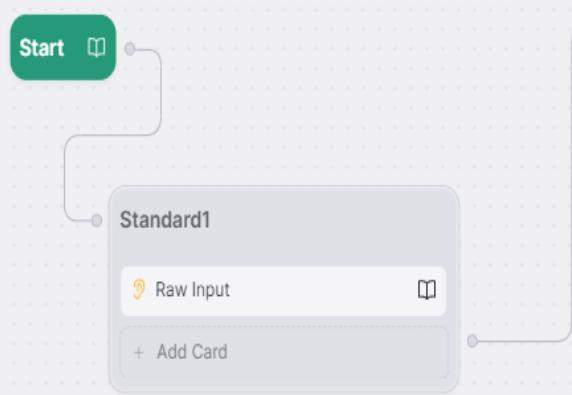


Figure 4.2.8: Usecase of customized chatbot

4.2 implementation Requirements

When considering the implementation requirements of a ProconnectEase system, several factors need to be taken into account. These requirements encompass technical, functional, and operational aspects to ensure a successful implementation. Here are some key implementation requirements to consider:

i. Hardware and Infrastructure:

Determine the hardware and infrastructure needed to support the ProconnectEase system. This may include servers, network infrastructure, storage devices, and other necessary equipment.

ii. Software and Technology Stack: Identify the appropriate software and technology stack required for the job management system. Consider the programming languages, frameworks, libraries, and databases that will be utilized.

iii. Database and Chatbot Use Case Design:

Design and implement a robust and efficient database structure for storing and managing data and managing the chatbot.

iv. Security and Access Control:

Implement strong security measures to protect sensitive data and prevent unauthorized access to the system. This may include user authentication, encryption, role-based access control, and other security mechanisms.

v. Integration:

Determine if the ProconnectEase system needs to integrate with other existing systems or external services, such as payment gateways, CRM systems, or email marketing platforms. Identify the necessary integration points and define the integration mechanisms.

vi. Data Backup and Recovery:

Implement regular data backup mechanisms to ensure the safety and integrity of the system's data. Define the frequency of backups and establish procedures for data recovery in case of system failures or data loss.

vii. Training and Documentation:

Provide training sessions and documentation for system administrators, staff members, and users to ensure they understand how to use the ProconnectEase system effectively. Create user manuals, system documentation, and any necessary training materials.

Chapter 5

Code

5.1 App.js:

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, Button } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

import { Login, Welcome, Signup, Jobs2, Job_des,
VideoProfile, Job_explore, Job_record, HomeScreen,
Profile, VideoRecord, Jobs3, Applyforum, ResumeForm,
ShowCV, JobPost, Addpostscreen, Chatbot } from './screens';

import { createDrawerNavigator } from '@react-navigation/drawer';
//import useAuth from './hooks/useAuth';
import { firebase } from './config/firebase';

import { useEffect, useState } from 'react';
import { LinearGradient } from 'expo-linear-gradient'

const Stack = createStackNavigator();
const Drawer = createDrawerNavigator();
export default function App() {
  const [initializing, setInitializing] = useState(true);
  const [user, setUser] = useState();
  //const { user } = useAuth();
  function onAuthStateChanged(user) {
    setUser(user);
    if (initializing) setInitializing(false);
  }

  useEffect(() => {
    const subscriber = firebase.auth().onAuthStateChanged(onAuthStateChanged);
  }
}
```

```

        return subscriber; // unsubscribe on unmount
    },[]);

if(initializing) return null;

if(user)
{
    return (
        <NavigationContainer>
        <Drawer.Navigator initialRouteName="HomeScreen">

            <Drawer.Screen name="HomeScreen" component={HomeScreen} />

            {/* <Drawer.Screen name="Jobs2" component={Jobs2} /> */}

            <Drawer.Screen name="Search Jobs" component={Jobs3} />

            <Drawer.Screen name="VideoProfile" component={VideoProfile} />

            <Drawer.Screen name="Profile" component={Profile} />

            <Drawer.Screen name="VideoRecord" component={VideoRecord} />

            <Drawer.Screen name="ResumeForm" component={ResumeForm} />

            <Drawer.Screen name="ShowCV" component={ShowCV} />

            <Drawer.Screen name="JobPost" component={JobPost} />

            <Drawer.Screen name="Applyforum" component={Applyforum} options={{ drawerLabel: () => null, // Hide the label drawerStyle: { display: 'none' }, // Hide the item in the drawer }} />

            <Drawer.Screen name="Job_des" component={Job_des} />
    );
}

```

```

        options={{
          drawerLabel: () => null, // Hide the label
          drawerStyle: { display: 'none' }, // Hide the item in the drawer
        }}
      />

<Drawer.Screen name="Addpostscreen" component={Addpostscreen}>
  options={{
    drawerLabel: () => null, // Hide the label
    drawerStyle: { display: 'none' }, // Hide the item in the drawer
  }}
</Drawer.Screen>

<Drawer.Screen name="Chatbot" component={Chatbot}>
  options={{
    drawerLabel: () => null, // Hide the label
    drawerStyle: { display: 'none' }, // Hide the item in the drawer
  }}
</Drawer.Screen>

 {/* <Drawer.Screen name="Logout" component={Logout} /> */}

</Drawer.Navigator>

</NavigationContainer>

);

}

else{
  return (

```

```

<NavigationContainer>
  <Stack.Navigator>
```

```

initialRouteName = "Welcome">

    <Stack.Screen name = "Welcome" component = { Welcome } options = { {headerShown: false} }/>

    <Stack.Screen name = "Login" component = { Login } options = { {headerShown: false} }/>

    <Stack.Screen name = "Signup" component = { Signup } options = { {headerShown: false} }/>

    {/* <Stack.Screen name = "Job_record" component = { Job_record } options = { {headerShown: false} }/> */}

</Stack.Navigator>
</NavigationContainer>

);

}

}

```

5.2 Index.js:

```

import Login from "./Login";

import Signup from "./Signup";

import Welcome from "./Welcome";

// import Jobs2 from "./Jobs2";

import Job_des from "./Job_des";

import VideoProfile from "./VideoProfile";

// import Job_explore from "./Job_explore";

import HomeScreen from "./HomeScreen";

import Profile from "./Profile";

```

```
import VideoRecord from "./VideoRecord";
import Jobs3 from "./Jobs3";
import Applyforum from "./Applyforum";
import ResumeForm from "./ResumeForm";
import ShowCV from "./ShowCV";
import JobPost from "./JobPost";
import Addpostscreen from "./Addpostscreen";
import Chatbot from "./Chatbot";
// import Job_record from "./Job_record";
export { Login, Signup, Welcome, Job_des, VideoProfile, HomeScreen, Profile, VideoRecord, Jobs3, Applyforum, ResumeForm, ShowCV, JobPost, Addpostscreen, Chatbot };
```

100% Polyester

5.3 AddPostScreen.js:

```
import React, { useState, useContext, useEffect } from 'react';
import { View, Text, Platform, StyleSheet, Alert,
ActivityIndicator, Button, TextInput, Image } from 'react-native';
import * as ImagePicker from 'expo-image-picker';
// import * as firebase from 'firebase';
import { firebase } from "../config.firebaseio";
import { StatusWrapper, SubmitBtn,
SubmitBtnText, InputWrapper, InputField } from "../styles/AddPost";
import { LinearGradient } from 'expo-linear-gradient'
```

```

;
import { ScrollView } from 'react-native-gesture-handler';

import { useNavigation } from "@react-navigation/native";

import * as FileSystem from 'expo-file-system';

const Addpostscreen = ({ route }) => {
  const { fetchPosts } = route.params;
  // const { user } = useContext(AuthContext);
  const navigation = useNavigation();

  const [currentUserName, setCurrentUserName] = useState("");
  const [profileimageuri, profilesetImageuri] = useState("");

  useEffect(() =>
    {
      firebase.firestore().collection('users')
        .doc(firebase.auth().currentUser.uid)
        .get()
        .then((documentSnapshot) => {
          if (documentSnapshot.exists) {
            setCurrentUserName(documentSnapshot.data().name);
            profilesetImageuri(documentSnapshot.data().imageuri);

          }
        })
        .else{
          console.log('User does not exist');
        }
    });
}

const currentUser = firebase.auth().currentUser;

const [image, setImage] = useState(null);
const [uploading, setUploading] = useState(false);
const [transferred, setTransferred] = useState(0);
const [post, setPost] = useState(null);

```

```
const takePhotoFromCamera = async () => {
  let result = await ImagePicker.launchCameraAsync({
    mediaTypes: ImagePicker.MediaTypeOptions.Images,
    allowsEditing: true,
    aspect: [4, 3],
    quality: 1,
  });

  handleImagePickerResult(result);
};
```

```
const choosePhotoFromLibrary = async () => {
  let result = await ImagePicker.launchImageLibraryAsync({
    mediaTypes: ImagePicker.MediaTypeOptions.Images,
    allowsEditing: true,
    aspect: [4, 3],
    quality: 1,
  });

  handleImagePickerResult(result);
};
```

```
const handleImagePickerResult = async (result) => {
  if (!result.canceled) {
    if (result.assets && result.assets.length > 0) {
      setImage(result.assets[0].uri);
    }
  }
};
```

```
const submitPost = async (fetchPostsCallback) => {
  if (!image) {
    Alert.alert('Error', 'Please select an image for your post.');
    return;
  }
```

```
if (!post) {
  Alert.alert('Error', 'Please write text for your post.');
  return;
}

const imageUrl = await uploadImage();

console.log('Image Url:', imageUrl);

console.log('Post:', post);

firebase.firestore().collection('posts').add({
  profileimg: profileimageuri,
  userName: currentUserName,
  userId: currentUserId,
  post: post,
  postImg: imageUrl,
  postTime: firebase.firestore.FieldValue.serverTimestamp(),
  likes: 0,
  comments: 0,
})

.then(() => {
  console.log('Post added!');
  Alert.alert(
    'Post published!',
    'Your post has been published Successfully!',
  );
  setPost(null);
  fetchPosts();
})

.catch((error) => {
  console.log('Something went wrong with added post to firestore.', error);
});

setImage(null);

};
```

```

const uploadImage = async () => {

  if( image == null ) {
    return null;
  }

  try{
    const {uri} = await FileSystem.getInfoAsync(image);

    const blob = await new Promise((resolve, reject) => {

      const xhr = new XMLHttpRequest();

      xhr.onload = function() {

        resolve(xhr.response);

      };

      xhr.onerror = function(e) {

        console.log(e);

        reject(new TypeError('Network request failed'));
      };

      xhr.responseType = 'blob';

      xhr.open('GET', uri, true);

      xhr.send(null);

    });

    setUploading(true);

    const filename = image.substring(image.lastIndexOf('/') + 1);
  }
}

```

```

const storageRef = firebase.storage().ref(`photos/${filename}`);
const task = storageRef.put(blob);

return new Promise((resolve, reject) => {
  task.on(
    'state_changed',
    (taskSnapshot) => {
      const progress =
        (taskSnapshot.bytesTransferred / taskSnapshot.totalBytes) * 100;
      console.log(`Upload is ${progress.toFixed(0)}% done`);

      setTransferred(progress.toFixed(0));
      // You can update the UI with the upload progress if needed
    },
    (error) => {
      console.log('Error uploading image:', error);
      reject(error); // Reject the promise if upload fails
    },
    async () => {
      // If upload is successful, get the download URL
      const downloadURL = await storageRef.getDownloadURL();
      console.log('Image uploaded successfully. Download URL:', downloadURL);
      setUploading(false);
      setImage(null);
      resolve(downloadURL); // Resolve the promise with the download URL
    }
  );
});

```

```
        } catch (error) {
            console.log('Error:', error);
        }

    };

return (

<View style={styles.container}>

    <Text>Select Image:</Text>

    <View style={styles.buttonContainer}>

        <View style={styles.button3}>
            <Button title="Go to previous screen" onPress={() => navigation.navigate("JobPost")} />
        </View>

        <View style={styles.button1}>
            <Button title="Take Photo" onPress={takePhotoFromCamera} />
        </View>

        <View style={styles.button2}>
            <Button title="Choose Photo" onPress={choosePhotoFromLibrary} />
        </View>

    </View>

);
```

```

<InputWrapper>
  {image && <Image source={{ uri: image }} style={{ width: 200, height: 200 }} />

<InputField
  placeholder="What's on your mind?"
  multiline
  numberOfLines={4}
  value={post}
  onChangeText={(content) => setPost(content)}
/>

{uploading ? (
  <StatusWrapper>
    <Text>{transferred} % Completed!</Text>
    <ActivityIndicator size="large" color="#0000ff" />
  </StatusWrapper>
) : (
  <SubmitBtn onPress={submitPost}>
    <SubmitBtnText>Post</SubmitBtnText>
  </SubmitBtn>
)
};

</InputWrapper>

</View>
);
};

```

```
export default Addpostscreen;

const styles = StyleSheet.create({

  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    paddingHorizontal: 20,
  },

  input: {
    borderWidth: 1,
    borderColor: '#ccc',
    borderRadius: 5,
    padding: 10,
    marginVertical: 10,
    width: '100%',
    height: 100,
  },

  progress: {
    alignItems: 'center',
    marginTop: 20,
  },

  buttonContainer: {
    marginTop: 30,
  },

  button1: {
    marginBottom: 20,
  },
})
```

```
        button3:{  
            marginBottom: 20,  
        },  
    );
```

5.4 ApplyForum.js:

```
import {  
    SafeAreaView,  
    TextInput,  
    Text,  
    TouchableOpacity,  
    View,  
    Button,  
    Linking  
} from "react-native";  
  
import React from "react";  
  
import Spacing from "../constants1/Spacing";  
  
import FontSize from "../constants1/FontSize";  
  
import Colors from "../constants1/Colors";  
  
import Font from "../constants1/Font";  
  
import { Ionicons } from "@expo/vector-icons";  
  
import { AntDesign } from '@expo/vector-icons';  
  
import {useState} from "react";  
  
import { useEffect} from "react";  
  
import * as MailComposer from "expo-mail-composer";  
  
import * as Print from "expo-print";  
  
import * as Permissions from 'expo-permissions';  
  
import { useNavigation } from "@react-navigation/native";  
  
const Applyforum = () => {  
    const navigation = useNavigation();
```

```

const [name, setName] = useState("");
const [focused, setFocused] = useState(false);

const [isAvailable, setIsAvailable] = useState(false);
const [recipients, setRecipients] = useState([]);
const [subject, setSubject] = useState(undefined);

const [body, setBody] = useState(undefined);
const [email, setEmail] = useState(undefined);

useEffect(() => {
  async function checkAvailability() {
    const isMailAvailable = await MailComposer.isAvailableAsync();
    setIsAvailable(isMailAvailable);
  }
  checkAvailability();
}, []);

// const sendMail = async () => {
//   const { uri } = await Print.printToFileAsync({
//     html: "<h1>My pdf!</h1>"
//   });
// }

// // const { status } = await Permissions.askAsync(Permissions.READ_CONTACTS);
// // if (status !== 'granted') {
// //   console.error('Contacts permission not granted');
// //   return;
// // }

```

```
// try {
//   MailComposer.composeAsync({
//     subject: subject,
//     body: body,
//     recipients: recipients,
//     attachments: [uri]
//   });
// } catch(error) {
//   console.log('Error sending email:', error); // Log the detailed error
// }
// 
```

```
const sendMail = () => {
  Linking.openURL(`mailto:${recipients.join(",")}?subject=${subject}&body=${body}`);
}
```

```
const addRecipient = () => {
  let newRecipients = [...recipients];
  newRecipients.push(email);
  setRecipients(newRecipients);
  setEmail(undefined);
};
```

```
const showRecipients = () => {
  if (recipients.length === 0) {
    return <Text>No recipients added</Text>;
```

```
}

return recipients.map((recipient, index) => {
  return <Text key={index}>{recipient}</Text>;
});

};

return (

<SafeAreaView>
  <View
    style={{{
      padding: Spacing * 2,
      backgroundColor: 'white',
      height: '100%',
    }}>

  <View
    style={{{
      alignItems: "center",
      marginTop: '20px',
    }}>

  <TouchableOpacity //133
    style={{{
      padding: Spacing,
      backgroundColor: Colors.gray,
      borderRadius: Spacing / 2,
      marginHorizontal: Spacing,
    }}>
```

```
    onPress={() => navigation.navigate("Search Jobs")}
  >

  <AntDesign name="arrowleft" size={24} color="black" />

</TouchableOpacity>

</View>

<View
  style={{

    marginVertical: Spacing * 3,
    backgroundColor: Colors.lightPrimary,
  }}
>

<TextInput
  onFocus={() => setFocused(true)}
  onBlur={() => setFocused(false)}
  placeholderTextColor={Colors.darkText}
  placeholder="Subject"
  value={subject}
  onChangeText= {setSubject}

  style={[
    {
      fontSize: FontSize.small,
      padding: Spacing * 2,
      backgroundColor: Colors.lightPrimary,
      borderRadius: Spacing,
      marginVertical: Spacing,
    }
  ]}
>
```

```
borderColor: 'black',
borderWidth: 1,
},
focused && {
borderWidth: 3,
borderColor: Colors.primary,
shadowOffset: { width: 4, height: Spacing },
shadowColor: Colors.primary,
shadowOpacity: 0.2,
shadowRadius: Spacing,
},
]}
]/>
```

```
<TextInput
  onFocus={() => setFocused(true)}
  onBlur={() => setFocused(false)}
  placeholderTextColor={Colors.darkText}
  placeholder="Message"
  value={body}
  onChangeText={setBody}
```

```
style={[
{
  fontSize: FontSize.small
```

```
,  
    padding: Spacing * 2,  
    backgroundColor: Colors.lightPrimary,  
    borderRadius: Spacing,  
    marginVertical: Spacing,  
    borderColor: 'black',  
    borderWidth: 1,  
,  
  
focused && {  
  borderWidth: 3,  
  borderColor: Colors.primary,  
  shadowOffset: { width: 4, height: Spacing },  
  shadowColor: Colors.primary,  
  shadowOpacity: 0.2,  
  shadowRadius: Spacing,  
,  
}  
]  
/>
```

```
<TextInput  
  onFocus={() => setFocused(true)}  
  onBlur={() => setFocused(false)}  
  placeholderTextColor={Colors.darkText}  
  placeholder="Email"  
  value={email}
```

```
onChangeText={setEmail}

style={[
  {
    fontSize: FontSize.small,
    padding: Spacing * 2,
    backgroundColor: Colors.lightPrimary,
    borderRadius: Spacing,
    marginVertical: Spacing,
    borderColor: 'black',
    borderWidth: 1,
  },
  focused && {
    borderWidth: 3,
    borderColor: Colors.primary,
    shadowOffset: { width: 4, height: Spacing },
    shadowColor: Colors.primary,
    shadowOpacity: 0.2,
    shadowRadius: Spacing,
  },
]}

/>

<Button title='Add Recipient' onPress={addRecipient}/>
{showRecipients()}
```

```
</View>

<TouchableOpacity
    style={{{
        padding: Spacing * 2,
        backgroundColor: Colors.primary,
        marginVertical: Spacing * 3,
        borderRadius: Spacing,
        shadowColor: Colors.primary,
        shadowOffset: {
            width: 0,
            height: Spacing,
        },
        shadowOpacity: 0.3,
        shadowRadius: Spacing,
    }}}

    onPress={() => sendMail()}
>

<Text
    style={{{
        color: Colors.onPrimary,
        textAlign: "center",
        fontSize: FontSize.large,
    }}}
>
    Send Email

```

```

        </Text>
    </TouchableOpacity>

<View
  style={{
    marginVertical: Spacing * 3,
  }}
>

<View
  style={{
    marginTop: Spacing,
    flexDirection: "row",
    justifyContent: "center",
  }}
>

</View>
</View>
</View>
</SafeAreaView>
);
};

export default Applyforum;

```

5.5 ChatBot.js:

```

import { StatusBar } from "expo-status-bar";
import { StyleSheet, Text, View } from "react-native";

```

```
import { WebView } from "react-native-webview";
// npx expo install react-native-webview

export default function Chatbot() {
  return (
    <WebView
      style={styles.container}
      source={{ uri: "https://mediafiles.botpress.cloud/96d614c1-dff8-463b-a258-24d76bab532d/webchat/bot.html" }}
    />
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#fff",
    alignItems: "center",
    justifyContent: "center",
    marginVertical: 64,
  },
});
```

5.6 HomeScreen.js:

```
import {createDrawerNavigator} from '@react-navigation/drawer';

import * as React from "react";

import { Image } from "expo-image";

import { StyleSheet, View, Text, Pressable, Alert, ScrollView } from "react-native";

import { LinearGradient } from "expo-linear-gradient";

import { Color, FontFamily, Border } from "../GlobalStyles1";

import * as Location from 'expo-location';

import * as IntentLauncher from 'expo-intent-launcher';

import { useNavigation } from "@react-navigation/native";

// import { Image } from 'react-native-expo-image-cache';

const Drawer = createDrawerNavigator();

const HomeScreen = () => {

    const navigation = useNavigation();

    const handlePress = async () => {

        // Check if location services are enabled

        let { status } = await Location.requestForegroundPermissionsAsync();

        if (status !== 'granted') {

            Alert.alert('Location permission not granted', 'Please enable location services to find
```

```

nearby hospitals.');

return;

}

// Get current location
let location = await Location.getCurrentPositionAsync({ });

const { latitude, longitude } = location.coords;

// Form Google Maps URL with current location
const mapsUrl =
`https://www.google.com/maps/search/software+firms+companies/@${latitude},${longitude}
,15z`;

// Open Google Maps or fallback to browser
try {

  await IntentLauncher.startActivityAsync('android.intent.action.VIEW', {
    data: mapsUrl,
  });

} catch (error) {

  console.error('Error opening Google Maps:', error);

  Alert.alert('Error', 'Failed to open Google Maps. Please try again later.');
};

}

return (
<ScrollView >

<View style={styles.onboarding}>

```

```
<View style={styles.frameParent}>

<Image
  style={[styles.groupChild, styles.groupChildPosition]}
  contentFit="cover"
  source={require("../assets/frame-27.png")}
/>

<Image
  style={[styles.groupItem, styles.groupLayout]}
  contentFit="cover"
  source={require("../assets/frame-28.png")}
/>

<Image
  style={styles.groupInner}
  contentFit="cover"
  source={require("../assets/frame-32.png")}
/>

<Image
  style={[styles.frameIcon, styles.frameIconLayout]}
  contentFit="cover"
  source={require("../assets/frame-31.png")}>
```

```
/>

<Image
  style={[styles.groupChild1, styles.groupChildPosition]}
  contentFit="cover"
  source={require("../assets/frame-33.png")}
/>

<Image
  style={styles.groupChild2}
  contentFit="cover"
  source={require("../assets/frame-24.png")}
/>

<Image
  style={ styles.groupChild3 }
  contentFit="cover"
  source={require("../assets/frame-321.png")}
/>

<Image
  style={ styles.groupChild4 }
  contentFit="cover"
  source={require("../assets/frame-34.png")}>
```

/>

```
<Image  
    style={ styles.groupChild5 }  
    contentFit="cover"  
    source={ require("../assets/frame-26.png") }  
/>
```

```
<Image  
    style={ styles.groupChild6 }  
    contentFit="cover"  
    source={ require("../assets/frame-311.png") }  
/>
```

```
<Image  
    style={[ styles.groupChild7, styles.childPosition ]}  
    contentFit="cover"  
    source={ require("../assets/frame-29.png") }  
/>
```

```
<Image  
    style={[ styles.groupChild8, styles.groupLayout ]}  
    contentFit="cover"
```

```
source={require("../assets/frame-331.png")}

/>>
```

```
<Image

style={styles.groupChild9}

contentFit="cover"

source={require("../assets/frame-312.png")}

/>>
```

```
<Image

style={styles.groupChild10}

contentFit="cover"

source={require("../assets/frame-30.png")}

/>>
```

```
<Image

style={[styles.groupChild11, styles.frameIconLayout]}

contentFit="cover"

source={require("../assets/frame-322.png")}

/>>
```

```
<Image

style={styles.groupChild12}

contentFit="cover"
```

```
source={require("../assets/frame-25.png")}

/>

<LinearGradient
  style={ styles.rectangleLineargradient }
  locations={[0, 1]}
  colors={"rgba(31, 31, 31, 0)", "#1f1f1f"}>
  />

</View>

<View style={ styles.yourSearchForTheNextDreamParent }>

  <Text
    style={[styles.yourSearchForContainer, styles.startSearchingTypo]}>
    </Text>

  <Text
    style={ styles.yourSearchFor }>
    >`Your search for the next dream job is over `</Text>

  <Text style={ styles.text }>🔗 </Text>
  </Text>

  <Pressable onPress={handlePress}>
    <View style={[styles.startSearchingParent, styles.frameChildFlexBox]}>
      <Text style={[styles.startSearching, styles.startSearchingTypo]}>
```

Nearby Software Companies

```
</Text>
```

```
{/* <Button
    title="Nearby Hospital"
    onPress={handlePress}
/> */
<Image
    style={styles.arrowRightIcon}
    contentFit="cover"
    source={require("../assets/arrowsright.png")}
/>
</View>

</Pressable>

<Pressable onPress={() => navigation.navigate("Chatbot")}>
    <View style={styles.imageContainer}>
        <Image style={styles.image} source= {require("../assets/download32.jpeg")}>
    </View>
</Pressable>
```

```
        <View style={[styles.frameChild, styles.frameChildFlexBox]} />
    </View>
</View>
</ScrollView>
);
};

const styles = StyleSheet.create({
    imageContainer: {
        marginLeft: 170,
        marginTop: 40,
    },
    image: {
        width: 150,
        height: 150,
        borderRadius: 80,
        marginBottom: 5,
    },
    groupChildPosition: {
        left: 7,
        position: "absolute",
    },
})
```

```
groupLayout: {  
    height: 76,  
    borderRadius: 39,  
    position: "absolute",  
},
```

```
frameIconLayout: {  
    width: 173,  
    position: "absolute",  
},
```

```
childPosition: {  
    left: 0,  
    position: "absolute",  
},
```

```
startSearchingTypo: {  
    textAlign: "center",  
    color: Color.colorWhite,  
    //  
    //fontFamily: FontFamily.satoshi,  
    fontWeight: "700",  
},
```

```
frameChildFlexBox: {  
    justifyContent: "center",  
    width: 340,  
    borderRadius: Border.br_13xl,
```

```
    alignItems: "center",  
},
```

```
groupChild: {  
  top: 379,  
  borderRadius: 47,  
  width: 185,  
  height: 92,  
},
```

```
groupItem: {  
  top: 396,  
  left: 196,  
  width: 175,  
},
```

```
groupInner: {  
  top: 316,  
  left: 371,  
  width: 135,  
  height: 165,  
  borderRadius: 39,  
  position: "absolute",  
},
```

```
frameIcon: {
```

```
top: 267,  
left: 230,  
height: 147,  
borderRadius: 39,  
,
```

```
groupChild1: {  
top: 298,  
width: 248,  
height: 90,  
borderRadius: 39,  
,
```

```
groupChild2: {  
top: 212,  
left: 38,  
borderRadius: 44,  
width: 189,  
height: 86,  
position: "absolute",  
,
```

```
groupChild3: {  
top: 205,  
left: 227,
```

```
width: 169,  
height: 87,  
borderRadius: 39,  
position: "absolute",  
,
```

```
groupChild4: {  
top: 231,  
left: 369,  
borderRadius: 33,  
width: 116,  
height: 121,  
position: "absolute",  
,
```

```
groupChild5: {  
top: 174,  
left: 320,  
borderRadius: 26,  
width: 138,  
height: 61,  
position: "absolute",  
,
```

```
groupChild6: {  
    top: 149,  
    left: 171,  
    borderRadius: 32,  
    width: 145,  
    height: 57,  
    position: "absolute",  
},
```

```
groupChild7: {  
    top: 116,  
    borderRadius: 43,  
    width: 171,  
    height: 117,  
},
```

```
groupChild8: {  
    top: 75,  
    left: 169,  
    width: 239,  
},
```

```
groupChild9: {  
    top: 34,  
    left: 404,  
    width: 76,
```

```
height: 148,  
borderRadius: 39,  
position: "absolute",  
,
```

```
groupChild10: {  
  
top: 6,  
left: 266,  
width: 148,  
height: 78,  
borderRadius: 39,  
position: "absolute",  
,
```

```
groupChild11: {  
  
top: 0,  
left: 91,  
borderRadius: 41,  
height: 82,  
,
```

```
groupChild12: {  
  
top: 78,  
left: 33,  
width: 117,
```

```
height: 62,  
position: "absolute",  
borderRadius: 24,  
,  
  
rectangleLineargradient: {  
top: 246,  
left: 51,  
width: 430,  
height: 243,  
backgroundColor: "transparent",  
position: "absolute",  
,  
  
frameParent: {  
top: 466,  
left: -51,  
width: 506,  
height: 489,  
position: "absolute",  
,  
  
yourSearchFor: {  
display: "flex",  
fontSize: 30,  
marginLeft: 16,
```

```
    },  
  
    text: {  
        fontSize: 32,  
    },  
  
    yourSearchForContainer: {  
        alignSelf: "stretch",  
        zIndex: 0,  
    },  
  
    startSearching: {  
        fontSize: 16,  
    },  
    arrowRightIcon: {  
        width: 24,  
        height: 24,  
        marginLeft: 8,  
    },  
  
    startSearchingParent: {  
        backgroundColor: "#5424fd",  
        flexDirection: "row",  
        padding: 16,  
        zIndex: 1,
```

```
marginTop: 40,  
},  
  
frameChild: {  
    top: 235,  
    backgroundColor: "rgba(84, 36, 253, 0.6)",  
    zIndex: 2,  
    left: 0,  
    position: "absolute",  
},  
  
yourSearchForTheNextDreamParent: {  
    top: 79,  
    left: 45,  
    alignItems: "center",  
    position: "absolute",  
},  
  
onboarding: {  
    backgroundColor: "#1f1f1f",  
    flex: 1,  
    width: "100%",  
    height: 932,
```

```
        overflow: "hidden",
        borderRadius: 0,
    },
});

export default HomeScreen;
```

5.7 ImageInput.js:

```
import React, { useEffect } from 'react';
import { View, Text, StyleSheet, TouchableWithoutFeedback, Alert, Image } from 'react-native';
import { MaterialCommunityIcons } from '@expo/vector-icons';
import * as ImagePicker from 'expo-image-picker';
//import { TouchableWithoutFeedback } from 'react-native-gesture-handler';
export default function ImageInput({imageuri, onChangeImage}) {
    useEffect(() => {
        requestpermission();
    }, [])
}

const requestpermission = async () => {
    const {granted} = await ImagePicker.requestMediaLibraryPermissionsAsync();
    if(!granted) alert('You need to enable permission to access the library');
```

```
    }

const handlepress = () => {
  if(!imageuri) {
    selectImage();
  }
}

else{
  Alert.alert('Delete', 'Are you sure you want to delete this image?['
    {text: 'Yes', onPress: () => onChangeImage(null)},
    {text: 'No'}
  ])
}
}
```

```
const selectImage = async () => {
  try {
    const result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      quality: 0.5,
    });

    if (!result.canceled) {
      onChangeImage(result.assets[0].uri); // Use result.assets[0].uri instead of result.uri
    }
  } catch (error) {
    console.log("Error reading an image", error);
  }
}
```

```

        }

    }

return (
    <TouchableWithoutFeedback onPress={handlepress}>
        <View style = {styles.container}>
            { !imageuri && <MaterialCommunityIcons color = 'darkgrey' name = 'camera' size =
{40} />}
            {imageuri && <Image source = { {uri: imageuri} } style = {styles.image}/>}
        </View>
    </TouchableWithoutFeedback>
)
}

const styles = StyleSheet.create({
    container: {
        backgroundColor: 'rgb(104,104,104)',
        justifyContent: 'center',
        alignItems: 'center',
        borderRadius: 15,
        overflow: 'hidden',
        height: 100,
        width: 100,
    },
},

```

```
image :  
{  
  width: '100%',  
  height: '100%',  
}  
})
```

5.8 Input.js:

```
import { View, Text, TextInput } from 'react-native'  
  
import React from 'react'  
  
const Input = ({title, placeholder, keyboard, is_password}) => {  
  
  return (  
    <View style={{marginVertical: 10}}>  
      <Text style={{fontSize: 16, color: '#03bafc'}}>{title}</Text>  
      <TextInput  
        placeholder={placeholder}  
        placeholderTextColor="gray"  
        style={  
          borderBottomColor: '#03bafc',  
          borderBottomWidth: 1,  
          paddingVertical: 0,  
          marginTop: 5,  
        }  
        >  
        secureTextEntry={is_password}  
      </TextInput>  
    </View>  
  )  
}
```

```

        keyboardType={keyboard}

    />
</View>

)

}

export default Input;

```

5.9 Job_des.js:

```

import * as React from "react";

import { Image, StyleSheet, Pressable, View, Text, Alert, FlatList, ScrollView, Button } from
"react-native";

//import LinearGradient from "react-native-linear-gradient";
//import { LinearGradient } from "expo-linear-gradient";

import { useNavigation } from "@react-navigation/native";

import { Color, Padding, FontSize, Border } from "../GlobalStyles";

import { useRoute } from "@react-navigation/native";

import * as Location from 'expo-location';

import * as IntentLauncher from 'expo-intent-launcher';

import { Linking } from 'react-native';

const Job_des = () => {

    const navigation = useNavigation();

    const route = useRoute();

```

```
try {
  const {
    job_name,
    company,
    salary,
    image,
    applybefore,
    jobtype,
    posted_on,
    job_description,
    job_qualityscore,
    job_applylink,
    longitiude,
    latitude,
    website,
  } = route.params;
```

```
const handleUrl = async () => {
  const url = await website;
  Linking.openURL(url);
}
```

```
const handlePress = async () => {
  // Check if location services are enabled
  let { status } = await Location.requestForegroundPermissionsAsync();
  if (status !== 'granted') {
    Alert.alert('Location permission not granted', 'Please enable location services to find
    nearby hospitals.');
  }
  return;
}
```

```

    // // Get current location
    // let location = await Location.getCurrentPositionAsync({ });

    // const { latitude, longitude } = location.coords;

    // Form Google Maps URL with current location

    console.log(latitude);

    console.log(longitude);

    const mapsUrl = `https://www.google.com/maps?q=${latitude},${longitude}`;

    // Open Google Maps or fallback to browser

    try {

        await IntentLauncher.startActivityAsync('android.intent.action.VIEW', {
            data: mapsUrl,
        })

    } catch (error) {

        console.error('Error opening Google Maps:', error);

        Alert.alert('Error', 'Failed to open Google Maps. Please try again later.');

    }

};

return (
    <View style={styles.screen}>
        <View style={[styles.frameParent, styles.parentPosition1]}>
            <ScrollView style={{

                backgroundColor: Color.primaryWHITE,
                maxHeight: 5000,

```

```

    } }>

    <View style={ styles.frameGroup }>

        <View style={[ styles.arrowBack24pxParent, styles.parentFlexBox ]}>

            <Pressable
                style={[ styles.arrowBack24px, styles.arrowBack24pxLayout ]}

                onPress={() => navigation.navigate("Jobs3")}
            >

                <Image
                    style={ styles.icon }
                    resizeMode="cover"

                    source={ require("../assets/arrow_back_24px.png") }
                />

            </Pressable>

            /* <Image
                style={[
                    styles.favoriteBorder24pxIcon,
                    styles.arrowBack24pxLayout,
                ]}
                resizeMode="cover"
                source={ require("../assets/favorite_border_24px.png") }
            /> */

        <Pressable onPress={() => navigation.navigate("Applyforum")}>

            <View style={ styles.applyNowWrapper }>

                <Text style={[ styles.applyNow, styles.applyTypo ]}>
                    APPLY NOW
                </Text>
            </View>
        </Pressable>
    </View>

```

```
</Text>

</View>

</Pressable>

</View>

<View style={styles.image3Parent}>

<Pressable onPress= {handleUrl}>

<Image
  style={styles.image3Icon}
  resizeMode="cover"
  source={{ uri: image }}
/>

</Pressable>

<View style={styles.midLevelUxDesignerParent}>

<Text style={[styles.midLevelUxDesigner, styles.toptalFlexBox]}>
  {job_name}
</Text>

<Text style={[styles.toptal, styles.toptalTypo]}>
  {company}
</Text>

<Text style={[styles.postedOn20, styles.july2021Typo]}>
  Posted on {posted_on}
</Text>

<Text>
  Job Quality Score: {job_qualityscore}
</Text>
```

```
</Text>

<Text>

    Job Apply Link: {job_applylink}

</Text>

</View>

</View>

</View>

<View style={[styles.frameWrapper, styles.frameWrapperSpaceBlock]}>

<View style={styles.frameContainer}>

    <View style={[styles.applyBeforeParent, styles.parentPosition1]}>

        <Text style={[styles.applyBefore, styles.applyTypo]}>

            Apply Before

        </Text>

        <Text style={[styles.july2021, styles.july2021Typo]}>

            {applybefore}

        </Text>

    </View>

    <View style={[styles.jobNatureParent, styles.parentPosition1]}>

        <Text style={[styles.applyBefore, styles.applyTypo]}>

            Job Nature

        </Text>

        <View style={styles.frameView}>

            <View style={styles.fullTimeWrapper}>

                <Text style={[styles.fullTime, styles.fullTimeLayout]}>

                    Concenture

                </Text>

            </View>

        </View>

    </View>

</View>
```

```
</Text>

</View>

<View style={styles.frameChild} />

</View>

</View>

<View style={[styles.salaryRangeParent, styles.parentPosition]}>

<Text style={[styles.applyBefore, styles.applyTypo]}>

    Job City

</Text>

<Text style={[styles.july2021, styles.july2021Typo]}>

    {salary}

</Text>

</View>

<View style={[styles.jobLocationParent, styles.parentPosition]}>

<Text style={[styles.applyBefore, styles.applyTypo]}>

    Job Type

</Text>

<Text style={[styles.july2021, styles.july2021Typo]}>

    {jobtype}

</Text>

</View>

</View>

</View>
```

```
<View style={[styles.frameWrapper1, styles.frameWrapperSpaceBlock]}>
```

```

<View style={styles.frameParent1}>

    <LinearGradient
        style={styles.jobDescriptionParent}
        locations={[0, 1]}
        colors={"#ffff", "rgba(255, 255, 255, 0)"}}
        useAngle={true}
        angle={180}
    >

    <Text style={[styles.applyBefore, styles.applyTypo]}>
        Job Description
    </Text>

    <Text style={[styles.canYouBring, styles.july2021Typo]}>
        {job_description}
    </Text>

    </LinearGradient>

</View>

</View>

</ScrollView>

/* <View style={[styles.frameWrapper2, styles.frameWrapperSpaceBlock]}>
<View>

    <View style={styles.applyNowWrapper}>
        <Text style={[styles.applyNow, styles.applyTypo]}>
            APPLY NOW
        </Text>
    </View>

```

```

        </View>
    </View>
    </View>
*)
</View>

</View>
);

}

catch (e)
{
    console.log(e);
    return Alert.alert(
        "No item selected to display",
        "Choose options",
        [
            {
                text: "Ask me later",
                onPress: () => console.log("Ask me later pressed"),
            },
            {
                text: "Cancel",
                onPress: () => console.log("Cancel Pressed"),
                style: "cancel",
            },
            {
                text: "OK", onPress: () => navigation.navigate("Jobs3")
            },
        ],
    );
}

```

```
{ cancelable: false }  
);  
}  
};  
  
const styles = StyleSheet.create({  
  
  parentPosition1: {  
    top: 0,  
    position: "absolute",  
  },  
  
  parentFlexBox: {  
    alignItems: "center",  
    flexDirection: "row",  
  },  
  
  arrowBack24pxLayout: {  
    height: 24,  
    width: 24,  
    position: "absolute",  
  },  
  
  toptalFlexBox: {  
    textAlign: "left",  
    color: Color.textColor,
```

```
},  
  
toptalTypo: {  
  
    opacity: 0.8,  
  
    //fontFamily: FontFamily.poppinsMedium,  
  
    fontWeight: "500",  
  
},  
  
july2021Typo: {  
  
    //fontFamily: FontFamily.poppinsRegular,  
  
    textAlign: "left",  
  
},  
  
frameWrapperSpaceBlock: {  
  
    padding: Padding.p_5xl,  
  
    marginTop: 10,  
  
    backgroundColor: Color.primaryWHITE,  
  
},  
  
applyTypo: {  
  
    //fontFamily: FontFamily.poppinsBold,  
  
    fontWeight: "700",  
  
    letterSpacing: 1,  
  
    textAlign: "left",
```

```
    },  
  
    fullTimeLayout: {  
        lineHeight: 13,  
        fontSize: FontSize.aPPLLabel12M_size,  
        color: Color.textColor,  
    },  
  
    parentPosition: {  
        top: 61,  
        position: "absolute",  
    },  
  
    seeTypo: {  
        color: Color.primary,  
        // fontFamily: FontFamily.poppinsSemiBold,  
        fontWeight: "600",  
        letterSpacing: 0,  
        textAlign: "left",  
    },  
  
    icon: {  
        height: "100%",  
        width: "100%"
```

```
,
```

```
    },
```



```
arrowBack24px: {
```

```
    left: -6,
```

```
    top: 27,
```

```
    zIndex: 0,
```

```
},
```

```
favoriteBorder24pxIcon: {
```

```
    top: 3,
```

```
    left: 339,
```

```
    zIndex: 1,
```

```
},
```



```
arrowBack24pxParent: {
```

```
    alignSelf: "stretch",
```

```
    justifyContent: "space-between",
```

```
},
```



```
image3Icon: {
```

```
    borderRadius: Border.br_9xs,
```

```
    width: 64,
```

```
    height: 64,
```

```
},
```

```
midLevelUxDesigner: {  
    fontSize: FontSize.size_7xl,  
    lineHeight: 31,  
    //fontFamily: fontFamily.poppinsMedium,  
    fontWeight: "500",  
    textAlign: "left",  
},  
  
toptal: {  
    marginTop: 6,  
    lineHeight: 15,  
    fontSize: FontSize.aPP14Medium_size,  
    textAlign: "left",  
    color: Color.textColor,  
},  
  
postedOn20: {  
    opacity: 0.6,  
    lineHeight: 13,  
    fontSize: FontSize.aPPLabel12M_size,  
    color: Color.textColor,  
    marginTop: 6,  
},
```

```
midLevelUxDesignerParent: {  
    marginTop: 16,  
},  
  
image3Parent: {  
    marginTop: 24,  
},  
  
frameGroup: {  
    paddingHorizontal: Padding.p_5xl,  
    paddingTop: Padding.p_11xl,  
    paddingBottom: Padding.p_xl,  
    width: 411,  
    backgroundColor: Color.primaryWHITE,  
},  
  
applyBefore: {  
    fontSize: FontSize.size_xs_5,  
    lineHeight: 12,  
    textTransform: "uppercase",  
    opacity: 0.75,  
    color: Color.textColor,  
},
```

```
july2021: {  
    marginTop: 10,  
    lineHeight: 15,  
    fontSize: FontSize.aPP14Medium_size,  
    color: Color.textColor,  
},  
applyBeforeParent: {  
    left: 0,  
},  
  
fullTime: {  
    textAlign: "right",  
    opacity: 0.8,  
    //fontFamily: fontFamily.poppinsMedium,  
    fontWeight: "500",  
},  
  
fullTimeWrapper: {  
    paddingHorizontal: Padding.p_5xs,  
    paddingVertical: Padding.p_9xs,  
    borderRadius: Border.br_6xl,  
    flexDirection: "row",  
    backgroundColor: Color.bG  
,  
},
```

```
frameChild: {  
    backgroundColor: Color.colorGhostwhite_100,  
    display: "none",  
    marginLeft: 8,  
    borderRadius: Border.br_6xl,  
},  
  
frameView: {  
    marginTop: 8,  
    flexDirection: "row",  
},  
  
jobNatureParent: {  
    left: 188,  
},  
  
salaryRangeParent: {  
    left: 0,  
},  
  
jobLocationParent: {  
    left: 188,  
},
```

```
frameContainer: {
```

```
    width: 334,
```

```
    height: 98,
```

```
},
```

```
frameWrapper: {
```

```
    marginTop: 10,
```

```
    width: 411,
```

```
},
```

```
canYouBring: {
```

```
    lineHeight: 20,
```

```
    color: Color.primaryBLACK,
```

```
    width: 363,
```

```
    marginTop: 12,
```

```
    fontSize: FontSize.aPP14Medium_size,
```

```
},
```

```
jobDescriptionParent: {
```

```
    backgroundColor: "transparent",  
},  
  
seeMore: {  
    fontSize: FontSize.size_smi,  
},  
  
arrowDropDown24px: {  
    width: 16,  
    height: 16,  
    marginLeft: 4,  
},  
  
seeMoreParent: {  
    marginTop: 16,  
},  
  
frameParent1: {  
    backgroundColor: Color.primaryWHITE,  
},  
  
frameWrapper1: {  
    marginTop: 10,  
},
```

```
seeMore1: {

    fontSize: FontSize.aPPLabel12M_size,
    color: Color.primary,
    //fontFamily: fontFamily.poppinsSemiBold,
    fontWeight: "600",
    letterSpacing: 0,
},

applyNow: {

    marginLeft: -45,
    top: 18,
    left: "50%",

    fontSize: FontSize.sFSubheadlineSemibold_size,
    color: Color.primaryWHITE,
    width: 97,
    height: 15,
    lineHeight: 15,
    position: "absolute",
    fontWeight: "700",
    letterSpacing: 1,
},

applyNowWrapper: {

    borderRadius: Border.br_xs,
```

```
backgroundColor: Color.primary,  
shadowColor: "rgba(28, 88, 242, 0.1)",
```

```
shadowOffset: {  
  width: 0,  
  height: 4,  
},
```

```
shadowRadius: 24,  
elevation: 24,  
shadowOpacity: 1,  
width: 250,  
height: 52,  
marginTop: 16,  
marginLeft: 90,
```

```
},
```

```
frameWrapper2: {  
  height: 175,  
  marginTop: 10,  
},
```

```
frameParent: {  
  width: 430,
```

```
    left: 0,  
    height: 943,  
},  
  
screen: {  
    flex: 1,  
    overflow: "hidden",  
    height: 943,  
    width: "100%",  
    // backgroundColor: 'black',  
},  
});  
  
export default Job_des;
```

5.11 Job_explore.js:

```
import * as React from "react";  
  
import { StyleSheet, View, Image, Text, Pressable } from "react-native";  
  
import { useNavigation } from "@react-navigation/native";  
  
import { Color, FontSize, Padding, Border } from "../GlobalStyles";  
  
const Job_explore = () => {  
    const navigation = useNavigation();
```

```

return (
  <View style={[styles.jobExplorJob2, styles.jobBg]}>
    <View style={[styles.component2, styles.component2Position]}>
      <View style={[styles.component2Child, styles.component2Position]} />
    </View>

    <View style={[styles.component59, styles.groupContainerLayout]}>
      <View style={[styles.component59Child, styles.bottomNavShadowBox]} />
      <View style={styles.component50Parent}>
        <Image
          style={[styles.component50Icon, styles.iconLayout1]}
          resizeMode="cover"
          source={require("../assets/Component50.png")}
        />

        <Text style={styles.myChats}>Company details</Text>

        <Image
          style={[styles.component50Icon1, styles.iconLayout1]}
          resizeMode="cover"
          source={require("../assets/Component50(1).png")}
        />

      </View>
    </View>
  <View style={[styles.us, styles.groupParentLayout]}>
    <View style={styles.usChild} />
    <View style={styles.groupParent}>

```

```
<View style={[styles.groupWrapper, styles.component2Position]}>

  <View style={[styles.component2, styles.component2Position]}>

    <Text style={[styles.uxProductEngineer, styles.samsungPosition]}>
      Ux Product Engineer
    </Text>

    <Text style={[styles.samsung, styles.samsungTypo]}>Samsung</Text>
  </View>

</View>

<Image
  style={[styles.component3Icon, styles.iconLayout1]}
  resizeMode="cover"
  source={require("../assets/Component3.png")}

/>

</View>

<View style={styles.frameParent}>

<View style={styles.frameGroup}>
```

```
<View style={styles.component50Group}>

    <Image
        style={styles.component50Icon2}
        resizeMode="cover"
        source={require("../assets/component-5021.png")}

    />

    <Text style={[styles.applications, styles.applicationsTypo]}>
        46 Applications
    </Text>

</View>

<View style={styles.frameWrapper}>

    <View style={styles.component50Group}>

        <Image
            style={styles.component50Icon2}
            resizeMode="cover"
            source={require("../assets/component-5031.png")}

        />

        <Text style={[styles.applications, styles.applicationsTypo]}>
            Applied
        </Text>
    
```

```
</Text>

</View>

</View>

<View style={[styles.instanceParent, styles.component2Position]}>

<View style={styles.internshipBorder}>

<Text style={[styles.internship, styles.internshipTypo]}>

    internship

</Text>

</View>

<View style={[styles.internshipContainer, styles.internshipBorder]}>

<Text style={[styles.internship, styles.internshipTypo]}>

    Permanent Job

</Text>

</View>

</View>

<View style={styles.viewApplicationParent}>

<Text style={styles.viewApplication}>View Application</Text>

<Image

    style={styles.component50Icon4}
```

```
    resizeMode="cover"

    source={require("../assets/component-5041.png")}

  />

</View>

<View style={styles.groupView}>

  <Image

    style={[styles.component50Icon5, styles.iconLayout]}

    resizeMode="cover"

    source={require("../assets/Component50(2).png")}

  />

  <Image

    style={[styles.component51Icon, styles.iconLayout]}

    resizeMode="cover"

    source={require("../assets/Component51.png")}

  />

</View>

</View>

<View style={[styles.groupContainer, styles.groupContainerLayout]}>

  <View style={[styles.frameContainer, styles.groupContainerLayout]}>

    <View style={[styles.rectangleParent, styles.rectangleFlexBox]}>

      <View style={[styles.frameChild, styles.frameLayout]} />
```

```
<View style={[styles.frameItem, styles.frameLayout]} />

<View style={[styles.frameInner, styles.frameLayout]} />

</View>

<View style={[styles.rectangleGroup, styles.rectangleFlexBox]}>

<Image

  style={styles.frameChild1Layout}

  resizeMode="cover"

  source={require("../assets/Rectangle18(1).png")}

/>

<Image

  style={[styles.frameChild1, styles.frameChild1Layout]}

  resizeMode="cover"

  source={require("../assets/Rectangle18(1).png")}

/>

<Image

  style={[styles.frameChild1, styles.frameChild1Layout]}

  resizeMode="cover"

  source={require("../assets/Rectangle18(1).png")}

/>

</View>

</View>
```

```
<View style={[styles.lineParent, styles.seeLayout]}>

  <View style={[styles.groupControl, styles.lineViewLayout]} />

  <View style={[styles.seeAll, styles.seeLayout]}>

    <Text style={[styles.exploreJobs, styles.samsungPosition]}>

      Company images

    </Text>

    <Text style={styles.seeTypo}>View More</Text>

  </View>

</View>

</View>

<View style={[styles.rectangleContainer, styles.groupControlLayout]}>

  <View style={[styles.groupControl, styles.groupControlLayout]} />

  <View style={[styles.groupParent1, styles.groupControlLayout]}>

    <View style={[styles.seeAll, styles.seeLayout]}>

      <View style={[styles.groupControl, styles.lineViewLayout]} />

      <View style={[styles.seeAll, styles.seeLayout]}>

        <Text style={[styles.exploreJobs, styles.samsungPosition]}>

          About company

        </Text>

        <Text style={[styles.seeAll3, styles.seeTypo]}>View More</Text>

      </View>

    </View>

  </View>

</View>
```

</View>

<Text style={[styles.samsungPayOffers, styles.internshipTypo]}>

Samsung Pay offers an online payment method which can be accessed from a web browser (PC or mobile). W3C mobile web payments. Merchant websites can have more, carry less. Samsung Pay gives you secure and easy access to your method of payment in more stores than any other mobile payment service.

</Text>

</View>

</View>

<View style={[styles.groupParent2, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineView, styles.lineViewLayout]} />

<View style={[styles.seeAll4, styles.seeLayout]}>

<Text style={[styles.exploreJobs, styles.samsungPosition]}>

About the internship

</Text>

<Text style={[styles.seeAll3, styles.seeTypo]}>View More</Text>

</View>

</View>

<Text style={[styles.setRealisticGoals, styles.applicationsTypo]}>

Set realistic goals for Your program

</Text>

<Text style={[styles.widenYourTalentContainer, styles.internshipTypo]}>

- {^1. Widen your talent pool. ...}
- 2. Look at the program from your interns' perspectives.
- 3. Get to know your interns. ...
- 4. Allow interns to get involved in company` }{" "}

work product. ...5. Encourage ongoing feedback. ...6. Offer guidance

based on your interns'needs.

</Text>

</View>

<View style={[styles.groupParent3, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineView, styles.lineViewLayout]} />

<View style={[styles.seeAll4, styles.seeLayout]}>

<Text style={[styles.exploreJobs, styles.samsungPosition]}>

Skills required

</Text>

<Text style={[styles.seeAll3, styles.seeTypo]}>View More</Text>

```
</View>

</View>

</View>

<Image

style={[styles.groupChild2, styles.groupChildLayout1]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild3, styles.groupChildLayout1]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild4, styles.groupChildPosition1]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild5, styles.groupChildPosition1]}
```

```
resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild6, styles.groupChildPosition]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild7, styles.groupChildPosition]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild8, styles.groupChildLayout1]}

resizeMode="cover"

source={require("../assets/Rectangle25.png")}

/>

<Image

style={[styles.groupChild9, styles.groupChildLayout1]}

resizeMode="cover"
```

```
source={require("../assets/Rectangle25.png")}

/>

</View>

<View style={[styles.groupParent4, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineContainer, styles.groupParentLayout]}>

<View style={[styles.lineView, styles.lineViewLayout]} />

<View style={[styles.seeAll4, styles.seeLayout]}>

<Text style={[styles.exploreJobs, styles.samsungPosition]}>

    Company requirement

</Text>

<Text style={[styles.seeAll3, styles.seeTypo]}>View More</Text>

</View>

</View>

</View>

<View style={[styles.rectangleParent1, styles.rectangleParentLayout]}>

<View style={[styles.rectangleView, styles.groupControlShadowBox]} />

<View
    style={[styles.rectangleParent2, styles.rectangleParentPosition]}

>

<Image
```

```
    style={[styles.groupChild11, styles.groupChildLayout]}

    resizeMode="cover"

    source={require("../assets/Rectangle25(1).png")}

  />

<Text

  style={[styles.certificate, styles.employeeTypo]}

>`Certificate`</Text>

<Text style={[styles.forAllSkills, styles.allTypo]}>

  For all skills

</Text>

</View>

</View>

<View style={[styles.rectangleParent3, styles.groupChild12Layout]}>

  <View style={[styles.groupChild12, styles.groupChild12Layout]} />

  <View

    style={[styles.rectangleParent4, styles.rectangleParentPosition]}

  >

    <Image

      style={[styles.groupChild13, styles.groupChildLayout]}

      resizeMode="cover"

      source={require("../assets/Rectangle25(1).png")}

    />
```

```
/>

<Text style={[styles.certificate, styles.employeeTypo]}>
    Job offer
</Text>

<Text style={[styles.allDetails, styles.allTypo]}>All Details</Text>

</View>

</View>

<View style={[styles.rectangleParent5, styles.groupControl14Layout]}>
    <View style={[styles.groupControl14, styles.groupControl14Layout]} />
    <View
        style={[styles.rectangleParent6, styles.rectangleParentPosition]}>
        >
        <Image
            style={[styles.groupControl15, styles.groupControlLayout]}
            resizeMode="cover"
            source={require("../assets/Rectangle25(1).png")}
        />
        <Text style={[styles.employee, styles.employeeTypo]}>Employee</Text>
        <Text style={[styles.yearsExperience, styles.allTypo]}>
            5 years experience
        </Text>
    </View>
</View>
```

```
</View>

</View>

</View>

<View style={[styles.bottomNav, styles.bottomNavShadowBox]}>

<View style={styles.tab}>

<Text style={[styles.icon, styles.iconFlexBox]} numberOfLines={1}>

</Text>

<Text style={[styles.title, styles.iconFlexBox]} numberOfLines={1}>
Home
</Text>

</View>

<Pressable
style={styles.tab}
onPress={() => navigation.navigate("Profile")}
>
<Text style={[styles.icon, styles.iconFlexBox]} numberOfLines={1}>

</Text>

<Text style={[styles.title, styles.iconFlexBox]} numberOfLines={1}>
```

Profile

```
</Text>

</Pressable>

<View style={styles.tab}>
```

```
    <Text style={[styles.icon, styles.iconFlexBox]} numberOfLines={1}>
```



```
    </Text>
```

```
    <Text style={[styles.title, styles.iconFlexBox]} numberOfLines={1}>
```

Jobs

```
    </Text>
```

```
</View>
```

```
<Pressable
```

```
    style={styles.tab}
```

```
    onPress={() => navigation.navigate("Frame")}
```

```
>
```

```
    <Text style={[styles.icon, styles.iconFlexBox]} numberOfLines={1}>
```



```
    </Text>
```

```
    <Text style={[styles.title, styles.iconFlexBox]} numberOfLines={1}>
```

Job Posting

```
</Text>

</Pressable>

<Pressable
  style={styles.tab}

  onPress={() => navigation.navigate("Search")}

>

<Text style={[styles.icon, styles.iconFlexBox]} numberOfLines={1}>

</Text>

<Text style={[styles.title, styles.iconFlexBox]} numberOfLines={1}>
  Search
</Text>

</Pressable>

</View>

</View>

);

};

const styles = StyleSheet.create({
  jobBg: {
    backgroundColor: Color.primaryWHITE,
```

```
width: "100%",  
},  
  
component2Position: {  
  
bottom: "0%",  
  
position: "absolute",  
},  
  
groupContainerLayout: {  
  
width: 375,  
  
left: 0,  
},  
  
bottomNavShadowBox: {  
  
shadowOpacity: 1,  
  
shadowOffset: {  
  
width: 0,  
  
height: 2,  
},  
  
shadowColor: "rgba(0, 0, 0, 0.12)",  
  
position: "absolute",  
},  
  
iconLayout1: {  
  
maxWidth: "100%",
```

```
maxHeight: "100%",

top: "0%",

position: "absolute",

overflow: "hidden",

},

groupParentLayout: {

width: 335,

position: "absolute",

},

samsungPosition: {

textAlign: "left",

left: "0%",

position: "absolute",

},

samsungTypo: {

//fontFamily: fontFamily.aPPTextButton14R,

color: Color.shadesOfBlackGrey3,

},

applicationsTypo: {

lineHeight: 16,

fontSize: FontSize.aPP14Medium_size,
```

```
    textAlign: "left",  
  
},  
  
internshipTypo: {  
  
    fontSize: FontSize.aPPLabel12M_size,  
  
    lineHeight: 16,  
  
    textAlign: "left",  
  
    //fontFamily: FontFamily.aPPM10,  
  
    fontWeight: "500",  
  
},  
  
internshipBorder: {  
  
    paddingVertical: Padding.p_9xs,  
  
    paddingHorizontal: Padding.p_5xs,  
  
    borderWidth: 0.5,  
  
    borderColor: Color.colorGainsboro_100,  
  
    borderRadius: Border.br_9xs,  
  
    justifyContent: "center",  
  
    flexDirection: "row",  
  
    alignItems: "center",  
  
    borderStyle: "solid",  
  
},  
  
iconLayout: {
```

```
width: "40%",  
  
maxWidth: "100%",  
  
maxHeight: "100%",  
  
bottom: "0%",  
  
top: "0%",  
  
height: "100%",  
  
position: "absolute",  
  
overflow: "hidden",  
  
},  
  
rectangleFlexBox: {  
  
justifyContent: "center",  
  
flexDirection: "row",  
  
alignItems: "center",  
  
position: "absolute",  
  
},  
  
frameLayout: {  
  
height: 2,  
  
width: 45,  
  
borderRadius: Border.br_11xs,  
  
},  
  
frameChild1Layout: {
```

```
height: 163,  
borderRadius: Border.br_sm,  
width: 335,  
,  
seeLayout: {  
height: 20,  
width: 335,  
position: "absolute",  
,  
lineViewLayout: {  
height: 0,  
borderTopWidth: 0.3,  
borderColor: Color.shadesOfBlackGrey3,  
width: 335,  
left: 0,  
borderStyle: "solid",  
position: "absolute",  
,  
groupItemLayout: {  
height: 164,  
width: 375,
```

```
    left: 0,  
  
    position: "absolute",  
  
},  
  
seeTypo: {  
  
    textAlign: "right",  
  
    left: "74.33%",  
  
    top: "27.5%",  
  
    width: "25.67%",  
  
    lineHeight: 16,  
  
    fontSize: FontSize.aPP14Medium_size,  
  
    color: Color.shadesOfBlackGrey3,  
  
    //fontFamily: FontFamily.aPPM10,  
  
    fontWeight: "500",  
  
    position: "absolute",  
  
},  
  
groupChildLayout1: {  
  
    height: 51,  
  
    width: 51,  
  
    position: "absolute",  
  
},  
  
groupChildPosition1: {
```

```
    left: 142,  
  
    height: 51,  
  
    width: 51,  
  
    position: "absolute",  
  
,  
  
groupChildPosition: {  
  
    left: 71,  
  
    height: 51,  
  
    width: 51,  
  
    position: "absolute",  
  
,  
  
rectangleParentLayout: {  
  
    height: 100,  
  
    top: 55,  
  
,  
  
groupChildShadowBox: {  
  
    height: 78,  
  
    shadowColor: "rgba(0, 0, 0, 0.1)",  
  
    borderRadius: Border.br_xs,  
  
    top: 22,  
  
    shadowOpacity: 1,
```

```
elevation: 3,  
shadowRadius: 3,  
shadowOffset: {  
  width: 0,  
  height: 2,  
},  
left: 0,  
backgroundColor: Color.primaryWHITE,  
},  
rectangleParentPosition: {  
  height: 88,  
  left: 8,  
  top: 0,  
  position: "absolute",  
},  
groupChildLayout: {  
  height: 44,  
  width: 44,  
  borderRadius: Border.br_81xl,  
  top: 0,  
  position: "absolute",
```

```
},  
  
employeeTypo: {  
  
    top: 52,  
  
    lineHeight: 16,  
  
    fontSize: FontSize.aPP14Medium_size,  
  
    textAlign: "left",  
  
    // fontFamily: FontFamily.aPPM10,  
  
    fontWeight: "500",  
  
    color: Color.primaryBLACK,  
  
    position: "absolute",  
  
},  
  
allTypo: {  
  
    color: Color.shadesOfBlackGrey2,  
  
    top: 72,  
  
    fontSize: FontSize.aPPM10_size,  
  
    lineHeight: 16,  
  
    //: FontFamily.aPPM10,  
  
    fontWeight: "500",  
  
    textAlign: "center",  
  
    position: "absolute",  
  
},
```

```
groupChild12Layout: {  
  
    width: 79,  
  
    position: "absolute",  
  
},  
  
groupChild14Layout: {  
  
    width: 110,  
  
    position: "absolute",  
  
},  
  
iconFlexBox: {  
  
    // fontFamily: FontFamily.robotoRegular,  
  
    justifyContent: "center",  
  
    alignItems: "center",  
  
    display: "flex",  
  
    textAlign: "center",  
  
    overflow: "hidden",  
  
},  
  
component2Child: {  
  
    left: "0%",  
  
    right: "0%",  
  
    top: "0%",  
  
    height: "100%",
```

```
bottom: "0%",  
  
width: "100%",  
  
backgroundColor: Color.primaryWHITE,  
  
},  
  
component2: {  
  
left: "0%",  
  
right: "0%",  
  
top: "0%",  
  
height: "100%",  
  
bottom: "0%",  
  
width: "100%",  
  
},  
  
border: {  
  
right: 2,  
  
borderRadius: 3,  
  
borderColor: Color.primaryBLACK,  
  
borderWidth: 1,  
  
width: 22,  
  
opacity: 0.35,  
  
borderStyle: "solid",  
  
bottom: "0%"},
```

```
    top: "0%",  
  
    height: "100%",  
  
    position: "absolute",  
  
},  
  
capIcon: {  
  
    height: "35.05%",  
  
    top: "31.96%",  
  
    bottom: "32.99%",  
  
    width: 1,  
  
    opacity: 0.4,  
  
    maxHeight: "100%",  
  
    right: 0,  
  
    position: "absolute",  
  
},  
  
capacity: {  
  
    height: "64.95%",  
  
    top: "17.53%",  
  
    right: 4,  
  
    bottom: "17.53%",  
  
    borderRadius: 1,  
  
    backgroundColor: Color.primaryBLACK,
```

```
width: 18,  
  
position: "absolute",  
  
},  
  
battery: {  
  
width: 24,  
  
right: 0,  
  
bottom: "0%",  
  
top: "0%",  
  
height: "100%",  
  
position: "absolute",  
  
},  
  
wifiIcon: {  
  
width: 15,  
  
height: 9,  
  
},  
  
cellularConnectionIcon: {  
  
width: 17,  
  
height: 9,  
  
},  
  
batteryParent: {  
  
height: "25.8%",
```

```
    top: "39.36%",  
  
    right: 14,  
  
    bottom: "34.84%",  
  
    width: 67,  
  
    position: "absolute",  
  
,  
  
time: {  
  
    marginTop: -5,  
  
    top: "50%",  
  
    fontSize: FontSize.sFSubheadlineSemibold_size,  
  
    letterSpacing: 0,  
  
    // fontFamily: FontFamily.sFSubheadlineSemibold,  
  
    textAlign: "center",  
  
    color: Color.primaryBLACK,  
  
    fontWeight: "600",  
  
    lineHeight: 20,  
  
    left: "0%",  
  
    position: "absolute",  
  
    width: "100%",  
  
,  
  
timeStyle: {
```

```
height: "47.87%",  
top: "15.96%",  
bottom: "36.17%",  
left: 21,  
width: 54,  
position: "absolute",  
},  
  
barsStatusBarDesignL: {  
height: "2.76%",  
bottom: "97.24%",  
left: "0%",  
right: "0%",  
top: "0%",  
position: "absolute",  
width: "100%",  
},  
  
component59Child: {  
elevation: 3,  
shadowRadius: 3,  
shadowOpacity: 1,  
shadowOffset: {
```

```
width: 0,  
  
height: 2,  
  
},  
  
shadowColor: "rgba(0, 0, 0, 0.12)",  
  
left: "0%",  
  
bottom: "0%",  
  
right: "0%",  
  
top: "0%",  
  
height: "100%",  
  
width: "100%",  
  
backgroundColor: Color.primaryWHITE,  
  
},  
  
component50Icon: {  
  
right: "92.84%",  
  
width: "7.16%",  
  
maxWidth: "100%",  
  
bottom: "0%",  
  
height: "100%",  
  
left: "0%",  
  
},  
  
myChats: {
```

```
width: "68.36%",  
left: "11.94%",  
alignItems: "center",  
display: "flex",  
textAlign: "left",  
//fontFamily: fontFamily.aPPM10,  
fontWeight: "500",  
lineHeight: 22,  
fontSize: FontSize.aPPTitle520M_size,  
color: Color.primaryBLACK,  
top: "0%",  
height: "100%",  
position: "absolute",  
},  
component50Icon1: {  
left: "92.84%",  
width: "7.16%",  
maxWidth: "100%",  
bottom: "0%",  
height: "100%",  
right: "0%",
```

```
},
```

```
component50Parent: {
```

```
height: "50%",
```

```
width: "89.33%",
```

```
top: "25%",
```

```
right: "5.33%",
```

```
bottom: "25%",
```

```
left: "5.33%",
```

```
position: "absolute",
```

```
},
```

```
component59: {
```

```
top: 44,
```

```
height: 48,
```

```
position: "absolute",
```

```
},
```

```
usChild: {
```

```
shadowRadius: 4,
```

```
elevation: 4,
```

```
borderRadius: Border.br_sm,
```

```
shadowOpacity: 1,
```

```
shadowOffset: {
```

```
width: 0,  
  
height: 2,  
  
},  
  
shadowColor: "rgba(0, 0, 0, 0.12)",  
  
left: "0%",  
  
bottom: "0%",  
  
right: "0%",  
  
top: "0%",  
  
height: "100%",  
  
position: "absolute",  
  
width: "100%",  
  
backgroundColor: Color.primaryWHITE,  
  
},  
  
uxProductEngineer: {  
  
//fontFamily: FontFamily.aPPM10,  
  
fontWeight: "500",  
  
lineHeight: 22,  
  
textAlign: "left",  
  
fontSize: FontSize.aPPTitle520M_size,  
  
color: Color.primaryBLACK,  
  
top: "0%",
```

```
width: "100%",  
},  
  
samsung: {  
  
width: "32.22%",  
  
top: "60%",  
  
fontSize: FontSize.aPPIInput16R_size,  
  
color: Color.shadesOfBlackGrey3,  
  
textAlign: "left",  
  
left: "0%",  
  
position: "absolute",  
  
lineHeight: 20,  
  
//fontFamily: FontFamily.aPPTextButton14R,  
},  
  
groupWrapper: {  
  
width: "81.02%",  
  
left: "18.98%",  
  
right: "0%",  
  
top: "0%",  
  
height: "100%",  
  
bottom: "0%",  
},
```

```
component3Icon: {
```

```
    height: "80%",
```

```
    width: "13.56%",
```

```
    right: "86.44%",
```

```
    bottom: "20%",
```

```
    left: "0%",
```

```
},
```

```
groupParent: {
```

```
    height: "23.81%",
```

```
    width: "88.06%",
```

```
    top: "11.43%",
```

```
    right: "5.97%",
```

```
    bottom: "64.76%",
```

```
    left: "5.97%",
```

```
    position: "absolute",
```

```
},
```

```
component50Icon2: {
```

```
    width: 16,
```

```
    height: 16,
```

```
    overflow: "hidden",
```

```
},
```

```
applications: {  
    marginLeft: 8,  
    color: Color.shadesOfBlackGrey3,  
    //fontFamily: FontFamily.aPPTextButton14R,  
},  
  
component50Group: {  
    flexDirection: "row",  
    alignItems: "center",  
},  
  
frameWrapper: {  
    marginLeft: 16,  
    flexDirection: "row",  
},  
  
frameGroup: {  
    height: "30.77%",  
    bottom: "69.23%",  
    flexDirection: "row",  
    left: 0,  
    top: "0%",  
    position: "absolute",  
},
```

```
internship: {  
    color: Color.shadesOfBlackGrey3,  
},  
  
internshipContainer: {  
    marginLeft: 16,  
},  
  
instanceParent: {  
    height: "46.15%",  
    width: "90.35%",  
    top: "53.85%",  
    right: "9.65%",  
    flexDirection: "row",  
    left: "0%",  
},  
  
frameParent: {  
    height: "24.76%",  
    top: "40.95%",  
    bottom: "34.29%",  
    width: 228,  
    left: 20,  
    position: "absolute",
```

```
    },  
  
    viewApplication: {  
  
        color: Color.primaryWHITE,  
  
        fontSize: FontSize.aPPLLabel12M_size,  
  
        lineHeight: 16,  
  
        textAlign: "left",  
  
        //fontFamily: FontFamily.aPPM10,  
  
        fontWeight: "500",  
  
    },  
  
    component50Icon4: {  
  
        height: 24,  
  
        marginLeft: 4,  
  
        width: 24,  
  
        overflow: "hidden",  
  
    },  
  
    viewApplicationParent: {  
  
        height: "15.24%",  
  
        width: "43.88%",  
  
        top: "73.33%",  
  
        right: "50.15%",  
  
        bottom: "11.43%",
```

```
backgroundColor: Color.successErrorBLUE,  
paddingVertical: Padding.p_9xs,  
paddingHorizontal: Padding.p_5xs,  
justifyContent: "center",  
borderRadius: Border.br_9xs,  
flexDirection: "row",  
left: "5.97%",  
alignItems: "center",  
position: "absolute",  
},  
  
component50Icon5: {  
right: "60%",  
left: "0%",  
},  
  
component51Icon: {  
left: "60%",  
right: "0%",  
},  
  
groupView: {  
height: "11.43%",  
width: "17.91%",
```

```
    top: "75.24%",  
  
    right: "3.58%",  
  
    bottom: "13.33%",  
  
    left: "78.51%",  
  
    position: "absolute",  
  
},  
  
us: {  
  
    top: 108,  
  
    height: 210,  
  
    left: 20,  
  
},  
  
frameChild: {  
  
    backgroundColor: Color.shadesOfBlackGrey5,  
  
    width: 45,  
  
    borderRadius: Border.br_11xs,  
  
},  
  
frameItem: {  
  
    width: 45,  
  
    borderRadius: Border.br_11xs,  
  
    backgroundColor: Color.successErrorBLUE,  
  
    marginLeft: 8,
```

```
    },  
  
    frameInner: {  
  
        backgroundColor: Color.shadesOfBlackGrey5,  
  
        width: 45,  
  
        borderRadius: Border.br_11xs,  
  
        marginLeft: 8,  
  
    },  
  
    rectangleParent: {  
  
        top: 175,  
  
        left: 157,  
  
        width: 61,  
  
    },  
  
    frameChild1: {  
  
        marginLeft: 20,  
  
    },  
  
    rectangleGroup: {  
  
        top: 0,  
  
        width: 375,  
  
        left: 0,  
  
    },  
  
    frameContainer: {
```

```
    top: 38,  
  
    height: 177,  
  
    position: "absolute",  
  
},  
  
groupChild: {  
  
    top: -16,  
  
    display: "none",  
  
},  
  
exploreJobs: {  
  
    width: "74.33%",  
  
    //fontFamily: FontFamily.aPPM10,  
  
    fontWeight: "500",  
  
    lineHeight: 22,  
  
    textAlign: "left",  
  
    fontSize: FontSize.aPPTitle520M_size,  
  
    color: Color.primaryBLACK,  
  
    top: "0%",  
  
},  
  
seeAll: {  
  
    top: 0,  
  
    left: 0,
```

```
    },  
  
    lineParent: {  
        top: 0,  
        left: 20,  
    },  
  
    groupContainer: {  
        top: 514,  
        height: 215,  
        position: "absolute",  
    },  
  
    groupItem: {  
        backgroundColor: Color.e1,  
        top: 0,  
    },  
  
    seeAll3: {  
        display: "none",  
    },  
  
    samsungPayOffers: {  
        top: 36,  
        width: 335,  
        position: "absolute",
```

```
    left: 0,  
  
    color: Color.primaryBLACK,  
  
    fontSize: FontSize.aPPLabel12M_size,  
  
},  
  
groupParent1: {  
  
    top: 16,  
  
    height: 132,  
  
    left: 20,  
  
},  
  
rectangleContainer: {  
  
    top: 334,  
  
},  
  
lineView: {  
  
    top: 0,  
  
},  
  
seeAll4: {  
  
    top: 19,  
  
    left: 0,  
  
},  
  
lineContainer: {  
  
    height: 39,
```

```
    top: 0,  
    left: 0,  
},  
  
setRealisticGoals: {  
  
    //fontFamily: FontFamily.aPPTextButton14SB,  
  
    color: Color.successErrorBLUE,  
  
    top: 55,  
    left: 0,  
  
    fontWeight: "600",  
  
    lineHeight: 16,  
  
    fontSize: FontSize.aPP14Medium_size,  
  
    position: "absolute",  
},  
  
widenYourTalentContainer: {  
  
    top: 79,  
    width: 335,  
  
    position: "absolute",  
  
    left: 0,  
  
    color: Color.primaryBLACK,  
  
    fontSize: FontSize.aPPLabel12M_size,  
},
```

```
groupParent2: {
```

```
    top: 745,
```

```
    height: 191,
```

```
    left: 20,
```

```
},
```

```
groupChild2: {
```

```
    top: 55,
```

```
    left: 0,
```

```
},
```

```
groupChild3: {
```

```
    top: 122,
```

```
    left: 0,
```

```
},
```

```
groupChild4: {
```

```
    top: 55,
```

```
},
```

```
groupChild5: {
```

```
    top: 122,
```

```
},
```

```
groupChild6: {
```

```
    top: 55,
```

},

groupChild7: {

top: 122,

},

groupChild8: {

left: 213,

top: 55,

},

groupChild9: {

left: 284,

top: 55,

},

groupParent3: {

top: 952,

height: 173,

left: 20,

},

rectangleView: {

width: 91,

position: "absolute",

},

```
groupChild11: {  
    left: 16,  
},  
  
certificate: {  
    left: 0,  
},  
  
forAllSkills: {  
    left: 8,  
    top: 72,  
},  
  
rectangleParent2: {  
    width: 75,  
},  
  
rectangleParent1: {  
    width: 91,  
    position: "absolute",  
    left: 0,  
},  
  
groupChild12: {  
    height: 78,  
    shadowColor: "rgba(0, 0, 0, 0.1)",
```

```
borderRadius: Border.br_xs,  
top: 22,  
shadowOpacity: 1,  
elevation: 3,  
shadowRadius: 3,  
shadowOffset: {  
  width: 0,  
  height: 2,  
},  
left: 0,  
backgroundColor: Color.primaryWHITE,  
},  
groupChild13: {  
  left: 10,  
},  
allDetails: {  
  left: 6,  
},  
rectangleParent4: {  
  width: 63,  
},
```

```
rectangleParent3: {  
    left: 111,  
    height: 100,  
    top: 55,  
},  
  
groupChild14: {  
    height: 78,  
    shadowColor: "rgba(0, 0, 0, 0.1)",  
    borderRadius: Border.br_xs,  
    top: 22,  
    shadowOpacity: 1,  
    elevation: 3,  
    shadowRadius: 3,  
    shadowOffset: {  
        width: 0,  
        height: 2,  
    },  
    left: 0,  
    backgroundColor: Color.primaryWHITE,  
},  
  
groupChild15: {
```

left: 25,

},

employee: {

left: 11,

},

yearsExperience: {

left: 0,

},

rectangleParent6: {

width: 94,

},

rectangleParent5: {

left: 210,

height: 100,

top: 55,

},

groupParent4: {

top: 1141,

height: 155,

left: 20,

},

```
icon: {  
    lineHeight: 28,  
    width: 28,  
    height: 28,  
    // fontFamily: FontFamily.robotoRegular,  
    fontSize: FontSize.aPPTitle520M_size,  
    color: Color.primaryBLACK,  
},  
  
title: {  
    alignSelf: "stretch",  
    lineHeight: 14,  
    height: 14,  
    fontSize: FontSize.aPPM10_size,  
    // fontFamily: FontFamily.robotoRegular,  
    color: Color.primaryWHITE,  
},  
  
tab: {  
    padding: Padding.p_9xs,  
    alignItems: "center",  
    flex: 1,  
},
```

```
bottomNav: {  
    top: 1541,  
    left: 2,  
    backgroundColor: Color.colorDarkslategray_100,  
    shadowRadius: 6,  
    elevation: 6,  
    width: 428,  
    flexDirection: "row",  
    shadowOpacity: 1,  
    shadowOffset: {  
        width: 0,  
        height: 2,  
    },  
    shadowColor: "rgba(0, 0, 0, 0.12)",  
    overflow: "hidden",  
},  
jobExplorJob2: {  
    height: 1361,  
    overflow: "hidden",  
    width: "100%",  
    flex: 1,
```

```
    },  
});  
  
export default Job_explore;
```

5.12Job_record.js:

```
import * as React from "react";  
  
import { Image, StyleSheet, View, Pressable, Text } from "react-native";  
  
import { useNavigation } from "@react-navigation/native";  
  
import { Color, Border, FontSize, Padding } from "../GlobalStyles";  
  
const Job_record = () => {  
  
  const navigation = useNavigation();  
  
  return (  
    <View style={styles.iphone14ProMax4}>  
      <Image  
        style={[styles.iphone14ProMax4Child, styles.iconBatteryLayout]}  
        resizeMode="cover"  
      />
```

```
source={require("../assets/group-14.png")}

/>

<View style={[styles.iphone14ProMax4Item, styles.iphone14Layout]} />

<View style={[styles.iphone14ProMax4Inner, styles.iphone14Layout]} />

<Pressable

  style={styles.icononlybulkarrowsLeftCircl}

  onPress={() => navigation.navigate("iPhone14ProMax2")}

>

<Image

  style={[styles.icon, styles.iconLayout1]}

  resizeMode="cover"

  source={require("../assets/icononlybulkarrowsleftcircle.png")}

/>

</Pressable>

<View style={styles.iphone678Se}>

<View style={[styles.modelight, styles.modedarkLayout]}>

<View style={[styles.rightSide, styles.sidePosition]}>

<Image

  style={[styles.iconBattery, styles.iconBatteryLayout]}

  resizeMode="cover"

  source={require("../assets/icon--battery.png")}
```

```
/>

<Text style={[styles.text, styles.textPosition]}>100%</Text>

<Image
  style={[styles.iconAlarm, styles.iconLayout]}
  resizeMode="cover"
  source={require("../assets/icon--alarm.png")}

/>

<Image
  style={[styles.iconBluetooth, styles.iconLayout]}
  resizeMode="cover"
  source={require("../assets/icon--bluetooth.png")}

/>

</View>

<View style={[styles.leftSide, styles.sidePosition]}>

<Image
  style={styles.iconSignal}
  resizeMode="cover"
  source={require("../assets/icon--signal.png")}

/>

<Text style={[styles.carrier, styles.carrierFlexBox]}>Figma</Text>

<Image
```

```
    style={styles.iconWifi}

    resizeMode="cover"

    source={require("../assets/icon--wifi.png")}

  />

</View>

</View>

<View style={[styles.modedark, styles.modedarkLayout]}>

<View style={[styles.iphone678Se1, styles.iphone678Se1Position]}>

<View style={[styles.rightSide, styles.sidePosition]}>

  <Image

    style={[styles.iconBattery, styles.iconBatteryLayout]}

    resizeMode="cover"

    source={require("../assets/icon--battery11.png")}

  />

  <Text style={[styles.text1, styles.text1Typo]}>100%</Text>

  <Image

    style={[styles.iconAlarm, styles.iconLayout]}

    resizeMode="cover"

    source={require("../assets/icon--alarm11.png")}

  />

  <Image
```

```
        style={[styles.iconBluetooth, styles.iconLayout]}

        resizeMode="cover"

        source={require("../assets/icon--bluetooth11.png")}

    />

</View>

<View style={[styles.leftSide, styles.sidePosition]}>

    <Image

        style={styles.iconSignal}

        resizeMode="cover"

        source={require("../assets/icon--signal11.png")}

    />

    <Text style={[styles.carrier1, styles.text1Typo]}>Figma</Text>

    <Image

        style={styles.iconWifi}

        resizeMode="cover"

        source={require("../assets/icon--wifi11.png")}

    />

</View>

</View>

</View>
```

```
<Text style={[styles.myProjects, styles.titleTypo3]}>My projects</Text>

<View style={[styles.rectangleView, styles.rectangleViewLayout]} />

<View
  style={[styles.iphone14ProMax4Child1, styles.rectangleViewLayout]}

/>

<Image
  style={[styles.vectorIcon, styles.vectorIconLayout1]}
  resizeMode="cover"
  source={require("../assets/vector1.png")}

/>

<Image
  style={[styles.vectorIcon1, styles.vectorIconLayout1]}
  resizeMode="cover"
  source={require("../assets/vector1.png")}

/>

<Image
  style={[styles.vectorIcon2, styles.vectorIconLayout]}
  resizeMode="cover"
  source={require("../assets/vector11.png")}

/>

<Image
```

```
style={[styles.vectorIcon3, styles.vectorIconLayout]}

resizeMode="cover"

source={require("../assets/vector11.png")}

/>


<View style={styles.ellipseParent}>

<Pressable

style={[styles.wrapper, styles.wrapperLayout]}

onPress={() => navigation.navigate("iPhone13ProMax1")}

>

<Image

style={styles.iconLayout2}

resizeMode="cover"

source={require("../assets/ellipse-4.png")}

/>

</Pressable>

<Image

style={[styles.icononlyboldcamera, styles.iconLayout1]}

resizeMode="cover"

source={require("../assets/icononlyboldcamera.png")}

/>

<Text style={[styles.title, styles.titleFlexBox]} numberOfLines={1}>
```

Record

</Text>

<Text style={[styles.title1, styles.titleFlexBox]} numberOfLines={1}>

Video 1

</Text>

<Text style={[styles.title2, styles.titleClr]} numberOfLines={1}>

19 Sep, 2023

</Text>

<Text style={[styles.title3, styles.titleTypo]} numberOfLines={1}>

01:06

</Text>

<Text style={[styles.title4, styles.titleTypo]} numberOfLines={1}>

01:06

</Text>

<Text style={[styles.title5, styles.titlePosition1]} numberOfLines={1}>

19 Sep, 2023

</Text>

<Text style={[styles.title6, styles.titlePosition1]} numberOfLines={1}>

Video 2

</Text>

<Text style={[styles.title7, styles.titleFlexBox]} numberOfLines={1}>

Show

</Text>

<Text style={styles.titlePosition} numberOfLines={1}>

Download

</Text>

<Text style={[styles.title9, styles.titlePosition]} numberOfLines={1}>

Download

</Text>

<Image

style={[styles.iconlyboldplay, styles.iphone678Se1Position]}

resizeMode="cover"

source={require("../assets/iconlyboldplay.png")}

/>

<Image

style={[styles.groupChild, styles.wrapperLayout]}

resizeMode="cover"

source={require("../assets/group-15.png")}

/>

</View>

<Pressable

style={[styles.rectangleParent, styles.groupItemLayout]}

```
        onPress={() => navigation.navigate("iPhone13ProMax1")}

    >

    <View style={[styles.groupItem, styles.groupItemLayout]} />

    <Text style={[styles.title10, styles.titleTypo1]} numberOfLines={1}>
        New Project
    </Text>

    <View style={[styles.camera, styles.cameraLayout]} />

    <Image
        style={styles.groupInner}
        resizeMode="cover"
        source={require("../assets/group-16.png")}
    />

    <Image
        style={[styles.plusIcon, styles.cameraLayout]}
        resizeMode="cover"
        source={require("../assets/plus.png")}
    />

    </Pressable>

</View>

);
```

```
const styles = StyleSheet.create({  
  iconBatteryLayout: {  
    height: 12,  
    position: "absolute",  
  },  
  
  iphone14Layout: {  
    height: 113,  
    width: 378,  
    borderWidth: 3,  
    borderColor: Color.primaryBLACK,  
    borderStyle: "solid",  
    borderRadius: Border.br_lgi,  
    left: 25,  
    position: "absolute",  
    backgroundColor: Color.colorSteelblue_100,  
  },  
  
  iconLayout1: {  
    maxHeight: "100%",  
    maxWidth: "100%",  
  },
```

```
modedarkLayout: {
```

```
    height: 20,
```

```
    width: 375,
```

```
},
```

```
sidePosition: {
```

```
    top: 3,
```

```
    position: "absolute",
```

```
},
```

```
textPosition: {
```

```
    textAlign: "right",
```

```
    right: 30,
```

```
    top: 0,
```

```
    position: "absolute",
```

```
},
```

```
iconLayout: {
```

```
    width: 14,
```

```
    height: 14,
```

```
    top: 0,
```

```
    position: "absolute",
```

```
},
```

```
carrierFlexBox: {
```

```
    textAlign: "left",  
  
    letterSpacing: 0,  
  
    marginLeft: 4,  
  
},  
  
iphone678Se1Position: {  
  
    top: "0%",  
  
    position: "absolute",  
  
    overflow: "hidden",  
  
},  
  
text1Typo: {  
  
    color: Color.primaryWHITE,  
  
    //fontFamily: FontFamily.sFSubheadlineSemibold,  
  
    lineHeight: 16,  
  
    fontSize: FontSize.aPPLabel12M_size,  
  
},  
  
titleTypo3: {  
  
    textAlign: "center",  
  
    // fontFamily: FontFamily.robotoRegular,  
  
    lineHeight: 12,  
  
    position: "absolute",  
  
},
```

```
rectangleViewLayout: {  
    height: 88,  
    width: 74,  
    backgroundColor: Color.colorGainsboro_300,  
    borderRadius: Border.br_mid,  
    position: "absolute",  
},  
  
vectorIconLayout1: {  
    width: "9.77%",  
    height: "3.76%",  
    maxHeight: "100%",  
    maxWidth: "100%",  
    position: "absolute",  
    overflow: "hidden",  
},  
  
vectorIconLayout: {  
    borderRadius: Border.br_114xl,  
    left: "80.47%",  
    right: "9.51%",  
    width: "10.02%",  
    height: "3.32%",
```

```
        maxHeight: "100%",

        maxWidth: "100%",

        position: "absolute",

        overflow: "hidden",

    },

    wrapperLayout: {

        height: 64,

        width: 68,

        top: 0,

        position: "absolute",

    },

    titleFlexBox: {

        justifyContent: "center",

        display: "flex",

        alignItems: "center",

    },

    titleClr: {

        color: Color.colorGray_500,

        fontSize: FontSize.size_smi,

    },

    titleTypo: {
```

```
fontSize: FontSize.aPPM10_size,  
color: Color.colorGray_500,  
textAlign: "center",  
//fontFamily: fontFamily.robotoRegular,  
lineHeight: 12,  
position: "absolute",  
overflow: "hidden",  
},  
  
titlePosition1: {  
left: 76,  
textAlign: "center",  
// fontFamily: fontFamily.robotoRegular,  
lineHeight: 12,  
position: "absolute",  
overflow: "hidden",  
},  
  
titlePosition: {  
left: 202,  
top: 86,  
fontSize: FontSize.aPPTitle520M_size,  
textAlign: "center",
```

```
// fontFamily: FontFamily.robotoRegular,  
  
lineHeight: 12,  
  
color: Color.primaryBLACK,  
  
position: "absolute",  
  
overflow: "hidden",  
  
},  
  
groupItemLayout: {  
  
height: 149,  
  
width: 378,  
  
position: "absolute",  
  
},  
  
titleTypo1: {  
  
fontSize: FontSize.aPPTitle520M_size,  
  
color: Color.primaryBLACK,  
  
},  
  
cameraLayout: {  
  
height: 24,  
  
width: 24,  
  
position: "absolute",  
  
overflow: "hidden",  
  
},
```

```
iphone14ProMax4Child: {
```

```
    top: 851,
```

```
    left: 167,
```

```
    width: 97,
```

```
},
```

```
iphone14ProMax4Item: {
```

```
    top: 466,
```

```
},
```

```
iphone14ProMax4Inner: {
```

```
    top: 622,
```

```
},
```

```
icon: {
```

```
    height: "100%",
```

```
    width: "100%",
```

```
    overflow: "hidden",
```

```
},
```

```
iconlybulkarrowLeftCircl: {
```

```
    left: "0.23%",
```

```
    top: "5.47%",
```

```
    right: "92.02%",
```

```
    bottom: "91.13%",
```

```
width: "7.74%",  
  
height: "3.4%",  
  
position: "absolute",  
  
},  
  
iconBattery: {  
  
top: 2,  
  
right: 0,  
  
width: 27,  
  
},  
  
text: {  
  
color: Color.primaryBLACK,  
  
// fontFamily: FontFamily.sFSubheadlineSemibold,  
  
lineHeight: 16,  
  
fontSize: FontSize.aPPLabel12M_size,  
  
},  
  
iconAlarm: {  
  
right: 77,  
  
height: 14,  
  
},  
  
iconBluetooth: {  
  
right: 62,
```

```
    height: 14,  
},  
  
rightSide: {  
  
    right: 6,  
  
    width: 91,  
  
    height: 16,  
  
},  
  
iconSignal: {  
  
    width: 21,  
  
    height: 14,  
  
},  
  
carrier: {  
  
    marginLeft: 4,  
  
    color: Color.primaryBLACK,  
  
    //fontFamily: FontFamily.sFSubheadlineSemibold,  
  
    lineHeight: 16,  
  
    fontSize: FontSize.aPPLabel12M_size,  
  
},  
  
iconWifi: {  
  
    width: 15,  
  
    marginLeft: 4,
```

```
height: 14,  
},  
  
leftSide: {  
  
    left: 4,  
  
    flexDirection: "row",  
  
    overflow: "hidden",  
  
},  
  
modelight: {  
  
    overflow: "hidden",  
  
},  
  
text1: {  
  
    textAlign: "right",  
  
    right: 30,  
  
    top: 0,  
  
    position: "absolute",  
  
},  
  
carrier1: {  
  
    marginLeft: 4,  
  
    textAlign: "left",  
  
    letterSpacing: 0,  
  
},
```

```
iphone678Se1: {  
  
    right: "0%",  
  
    bottom: "0%",  
  
    left: "0%",  
  
    height: "100%",  
  
    width: "100%",  
  
},  
  
modedark: {  
  
    marginTop: 32,  
  
},  
  
iphone678Se: {  
  
    borderRadius: 5,  
  
    borderStyle: "dashed",  
  
    borderColor: "#333",  
  
    borderWidth: 1,  
  
    width: 430,  
  
    height: 47,  
  
    padding: Padding.p_base,  
  
    alignItems: "center",  
  
    left: 0,  
  
    top: 0,
```

```
position: "absolute",
overflow: "hidden",
backgroundColor: Color.colorSteelblue_100,
},
myProjects: {
    top: 411,
    left: 28,
    fontSize: 24,
    color: Color.primaryBLACK,
},
rectangleView: {
    top: 479,
    left: 49,
},
iphone14ProMax4Child1: {
    top: 635,
    left: 46,
},
vectorIcon: {
    top: "54.18%",
    right: "75.12%",
```

bottom: "42.06%",

left: "15.12%",

},

vectorIcon1: {

top: "70.92%",

right: "75.81%",

bottom: "25.32%",

left: "14.42%",

},

vectorIcon2: {

top: "50.86%",

bottom: "45.83%",

},

vectorIcon3: {

top: "67.6%",

bottom: "29.09%",

},

iconLayout2: {

height: "100%",

width: "100%",

},

```
wrapper: {  
    left: 0,  
},  
  
iconlyboldcamera: {  
    height: "3.39%",  
    width: "7.88%",  
    top: "3.55%",  
    right: "84.59%",  
    bottom: "93.05%",  
    left: "7.53%",  
    position: "absolute",  
    overflow: "hidden",  
},  
  
title: {  
    fontSize: FontSize.aPPTitle520M_size,  
    color: Color.primaryBLACK,  
    top: 85,  
    justifyContent: "center",  
    display: "flex",  
    width: 68,  
    textAlign: "center",
```

```
//fontFamily: FontFamily.robotoRegular,  
  
lineHeight: 12,  
  
height: 14,  
  
position: "absolute",  
  
overflow: "hidden",  
  
left: 0,  
  
},  
  
title1: {  
  
top: 391,  
  
left: 68,  
  
fontSize: FontSize.aPPTitle520M_size,  
  
color: Color.primaryBLACK,  
  
width: 68,  
  
justifyContent: "center",  
  
display: "flex",  
  
height: 14,  
  
textAlign: "center",  
  
//fontFamily: FontFamily.robotoRegular,  
  
lineHeight: 12,
```

position: "absolute",

overflow: "hidden",

```
    },  
  
    title2: {  
  
        top: 426,  
  
        left: 72,  
  
        textAlign: "center",  
  
        //fontFamily: FontFamily.robotoRegular,  
  
        lineHeight: 12,  
  
        position: "absolute",  
  
        overflow: "hidden",  
  
    },  
  
    title3: {  
  
        top: 448,  
  
        left: 72,  
  
    },  
  
    title4: {  
  
        top: 607,  
  
        left: 77,  
  
    },  
  
    title5: {  
  
        top: 585,  
  
        color: Color.colorGray_500,
```

```
fontSize: FontSize.size_smi,  
},  
  
title6: {  
  
  top: 550,  
  
  justifyContent: "center",  
  
  display: "flex",  
  
  alignItems: "center",  
  
  fontSize: FontSize.aPPTitle520M_size,  
  
  color: Color.primaryBLACK,  
  
  width: 68,  
  
  height: 14,  
  
},  
  
title7: {  
  
  left: 107,  
  
  fontSize: FontSize.aPPTitle520M_size,  
  
  color: Color.primaryBLACK,  
  
  top: 85,  
  
  justifyContent: "center",  
  
  display: "flex",  
  
  width: 68,  
  
  textAlign: "center",
```

```
//fontFamily: FontFamily.robotoRegular,  
  
lineHeight: 12,  
  
height: 14,  
  
position: "absolute",  
  
overflow: "hidden",  
  
},  
  
title9: {  
  
width: 90,  
  
height: 13,  
  
justifyContent: "center",  
  
display: "flex",  
  
alignItems: "center",  
  
},  
  
iconlyboldplay: {  
  
height: "10.34%",  
  
width: "23.29%",  
  
right: "40.07%",  
  
bottom: "89.66%",  
  
left: "36.64%",  
  
maxHeight: "100%",  
  
maxWidth: "100%",
```

```
    },  
  
    groupChild: {  
  
        left: 214,  
  
    },  
  
    ellipseParent: {  
  
        top: 96,  
  
        left: 69,  
  
        width: 292,  
  
        height: 619,  
  
        position: "absolute",  
  
    },  
  
    groupItem: {  
  
        left: 0,  
  
        top: 0,  
  
        height: 149,  
  
        borderWidth: 3,  
  
        borderColor: Color.primaryBLACK,  
  
        borderStyle: "solid",  
  
        borderRadius: Border.br_lgi,  
  
        backgroundColor: Color.colorSteelblue_100,  
  
    },
```

```
title10: {  
    top: 94,  
    left: 135,  
    textAlign: "center",  
    //fontFamily: FontFamily.robotoRegular,  
    lineHeight: 12,  
    position: "absolute",  
    overflow: "hidden",  
},  
  
camera: {  
    top: 124,  
    left: 291,  
},  
  
groupInner: {  
    top: 28,  
    left: 164,  
    width: 49,  
    height: 45,  
    position: "absolute",  
},  
  
plusIcon: {
```

```
top: 39,  
left: 177,  
},  
rectangleParent: {  
    top: 231,  
    left: 26,  
},  
iphone14ProMax4: {  
    flex: 1,  
    height: 932,  
    overflow: "hidden",  
    width: "100%",  
    backgroundColor: Color.colorSteelblue_100,  
},  
});  
  
export default Job_record;
```

5.13 JobPost.js:

```
// import React, { useEffect, useState } from "react";  
  
// import {  
  
//   FlatList,  
  
//   Button,  
  
// } from "react-native";  
  
  
  
  
// import { useNavigation } from "@react-navigation/native";  
  
// import { firebase } from "../config.firebaseio";  
  
// import {  
  
//   Card,  
  
//   Container,  
  
//   Divider,  
  
//   Interaction,  
  
//   InteractionText,  
  
//   InteractionWrapper,  
  
//   PostText,  
  
//   UserImg,  
  
//   UserInfo,  
  
//   UserInfoText,  
  
//   UserName,  
  
//   PostTime,  
// }
```

```
// PostImg,  
  
// } from "../styles/FeedStyles";  
  
// import {Ionicons} from "react-native-vector-icons";  
  
// const Posts = [  
  
// {  
  
// id: "1",  
  
// userName: "Jenny Doe",  
  
// userImg: require("../assets/users/user-3.jpg"),  
  
// postTime: "4 mins ago",  
  
// post: "Hey there, this is my test for a post of my social app in React Native.",  
  
// postImg: require("../assets/posts/post-img-3.jpg"),  
  
// liked: true,  
  
// likes: "14",  
  
// comments: "5",  
  
// },  
  
// {  
  
// id: "2",  
  
// userName: "John Doe",  
  
// userImg: require("../assets/users/user-1.jpg"),  
  
// postTime: "2 hours ago",
```

```
// post: "Hey there, this is my test for a post of my social app in React Native.",  
  
// postImg: "none",  
  
// liked: false,  
  
// likes: "8",  
  
// comments: "0",  
  
// },  
  
// {  
  
// id: "3",  
  
// userName: "Ken William",  
  
// userImg: require("../assets/users/user-4.jpg"),  
  
// postTime: "1 hours ago",  
  
// post: "Hey there, this is my test for a post of my social app in React Native.",  
  
// postImg: require("../assets/posts/post-img-2.jpg"),  
  
// liked: true,  
  
// likes: "1",  
  
// comments: "0",  
  
// },  
  
// {  
  
// id: "4",  
  
// userName: "Selina Paul",  
  
// userImg: require("../assets/users/user-6.jpg"),
```

```
//  postTime: "1 day ago",  
  
//  post: "Hey there, this is my test for a post of my social app in React Native.",  
  
//  postImg: require("../assets/posts/post-img-4.jpg"),  
  
//  liked: true,  
  
//  likes: "22",  
  
//  comments: "4",  
  
// },  
  
// {  
  
//  id: "5",  
  
//  userName: "Christy Alex",  
  
//  userImg: require("../assets/users/user-7.jpg"),  
  
//  postTime: "2 days ago",  
  
//  post: "Hey there, this is my test for a post of my social app in React Native.",  
  
//  postImg: "none",  
  
//  liked: false,  
  
//  likes: "0",  
  
//  comments: "0",  
  
// },  
// ];
```

```
// export default function JobPost() {  
  
//   const fetchPosts = async () => {  
  
//     try {  
  
//       const list = [];  
  
//       await firebase.firestore()  
  
//         .collection('posts')  
  
//         .orderBy('postTime', 'desc')  
  
//         .get()  
  
//       .then((querySnapshot) => {  
  
//         console.log('Total Posts: ', querySnapshot.size);  
  
//         querySnapshot.forEach((doc) => {  
  
//           const {  
//             userId,  
//             post,  
//             postImg,  
//             postTime,  
//             likes,  
//             comments,  
//           } = doc.data();  
//           list.push({  
//             id: doc.id,  
//             userId,  
//             post,  
//             postImg,  
//             postTime,  
//             likes,  
//             comments,  
//           });  
//         });  
//       } catch (error) {  
//         console.error(error);  
//       }  
//     }  
//   };  
// };
```

```
//      } = doc.data();

//      list.push({

//      id: doc.id,

//      userId,

//      userName: 'Test Name',

//      userImg:

//      'https://lh5.googleusercontent.com/-

b0PKyNuQv5s/AAAAAAAAAAI/AAAAAAAAM/AMZuuclxAM4M1SCBGAO7Rp-

QP6zgBEUkOQ/s96-c/photo.jpg',

//      postTime: postTime,

//      post,

//      postImg,

//      liked: false,

//      likes,

//      comments,

//      });

//      });

//      });

//      setPosts(list);
```

```
//     if (loading) {  
  
//         setLoading(false);  
  
//     }  
  
//     console.log('Posts: ', posts);  
  
// } catch (e) {  
  
//     console.log(e);  
  
// }  
  
// };  
  
// useEffect(() => {  
  
//     fetchPosts();  
  
// }, []);  
  
// const navigation = useNavigation();  
  
// const renderPostItem = ({ item }) => {
```

```
// let likeIconColor = item.liked ? 'red' : 'black';

// let likeText, commentText;

// if (item.likes == 1) {

//   likeText = '1 Like';

// } else if (item.likes > 1) {

//   likeText = item.likes + ' Likes';

// } else {

//   likeText = 'Like';

// }

// if (item.comments == 1) {

//   commentText = '1 Comment';

// } else if (item.comments > 1) {

//   commentText = item.comments + ' Comments';

// } else {

//   commentText = 'Comment';

// }

// return (

//   <Card>
```

```
//      <UserInfo>

//      <UserImg source={item.userImg} />

//      <UserInfoText>

//      <UserName>{ item.userName }</UserName>

//      <PostTime>{ item.postTime }</PostTime>

//      </UserInfoText>

//      </UserInfo>

//      <PostText>{item.post}</PostText>

//      {item.postImg !== "none" ? (

//      <PostImg source={item.postImg} />

//      ) : (

//      <Divider />

//      )}

//      <Divider />

//      <InteractionWrapper>

//      <Interaction active={item.liked}>

//      <Ionicons name={likeIcon} size={25} color={likeIconColor} />
```

```

//      <InteractionText active={item.liked}>{likeText}</InteractionText>

//      </Interaction>

//      <Interaction>

//      <Ionicons name="md-chatbubble-outline" size={25} color="red" />

//      <InteractionText>{commentText}</InteractionText>

//      </Interaction>

//      </InteractionWrapper>

//      </Card>

//      );

//  };



//  return (


//  <Container>

//  <Button

//  title="Post a Job"

//  onPress={() => navigation.navigate('Addpostscreen')}

//  />

//  <FlatList

//  data={Posts}

//  renderItem={renderPostItem}

```

```
//      keyExtractor={(item) => item.id}

//      showsVerticalScrollIndicator={false}

//    />

//  </Container>

// );

// }
```

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

```
import {

  View,
  ScrollView,
  Text,
  StyleSheet,
  FlatList,
  SafeAreaView,
  Alert,
  Button,
```

ActivityIndicator

Image,

`TouchableOpacity`,

Share

```
} from 'react-native';
```

```
// import SkeletonPlaceholder from 'react-native-skeleton-placeholder';
```

```
import {LinearGradient} from 'expo-linear-gradient';
```

```
import { useNavigation } from "@react-navigation/native";
```

```
import {Ionicons} from "react-native-vector-icons";
```

```
import {
```

Card,

Container,

Divider,

Interaction,

InteractionText,

InteractionWrapper,

PostText,

```
UserImg,  
UserInfo,  
UserInfoText,  
UserName,  
PostTime,  
PostImg,  
  
 } from "../styles/FeedStyles";  
  
import {firebase} from "../config.firebaseio";  
  
import moment from 'moment';  
  
// import * as Sharing from 'expo-sharing';  
// import * as FileSystem from 'expo-file-system';  
  
const JobPost = () => {  
  
  const navigation = useNavigation();
```

```
const [posts, setPosts] = useState(null);

const [loading, setLoading] = useState(false);

const [deleted, setDeleted] = useState(false);

const currentUserId = firebase.auth().currentUser.uid;
```

```
const renderPostItem = ({ item }) => {

  let likeIcon = item.liked ? 'heart' : 'heart-outline';

  let likeIconColor = item.liked ? 'red' : 'black';

  let likeText, commentText;

  if (item.likes == 1) {

    likeText = '1 Like';

  } else if (item.likes > 1) {

    likeText = item.likes + ' Likes';

  } else {

    likeText = 'Like';

  }

}
```

```
if (item.comments == 1) {  
  
    commentText = '1 Comment';  
  
} else if (item.comments > 1) {  
  
    commentText = item.comments + ' Comments';  
  
} else {  
  
    commentText = 'Comment';  
  
}
```

```
const url = item.postImg;  
  
const handleShare = async () => {  
  
    try {  
  
        const result = await Share.share({  
  
            message:  
                ('Check out the post'+ '\n'+ url )  
  
        });  
  
        if (result.action === Share.sharedAction) {
```

```
if (result.activityType) {  
    // shared with activity type of result.activityType  
  
} else {  
    // shared  
  
}  
  
} else if (result.action === Share.dismissedAction) {  
  
    // dismissed  
  
}  
  
} catch (error) {  
  
    alert(error.message);  
  
}  
  
};  
  
return (
```

```
<View style={styles.card}>

<View style={styles.userInfo}>

  <Image source={{ uri: item.userImg }} style={styles.userImg} />

  <View style={styles.userInfoText}>

    <Text style={styles.userName}>{item.userName}</Text>

    <Text style={styles.postTime}>{moment(item.postTime.toDate()).fromNow()}</Text>

  </View>

</View>

<Text style={styles.postText}>{item.post}</Text>

{item.postImg !== null ? <Image source={{ uri: item.postImg }} style={styles.postImg} /> : <View style={styles.divider} /> }

<View style={styles.divider} />

<View style={styles.interactionWrapper}>

  <TouchableOpacity style={[styles.interaction, item.liked && styles.activeInteraction]}>

    <Ionicons name={likeIcon} size={25} color={likeIconColor} />

    <Text style={[styles.interactionText, item.liked && styles.activeInteractionText]}>{likeText}</Text>

  </TouchableOpacity>

  <TouchableOpacity style={styles.interaction}>

    onPress={handleShare}

  </TouchableOpacity>

```

```
// onPress={() => handleShare('https://bauet.ac.bd/', 'Check out this post!')}
```

>

```
<Ionicons name="share-social-outline" size={25} color="red" />
```

```
<Text style={styles.interactionText}>Share</Text>
```

```
</TouchableOpacity>
```

```
{currentUserId === item.userId && (
```

```
<TouchableOpacity style={styles.interaction} onPress={handleDelete}>
```

```
<Ionicons name="md-trash-bin" size={25} color="red" />
```

```
</TouchableOpacity>
```

```
)}
```

```
</View>
```

```
</View>
```

```
);
```

```
};
```

```
const fetchPosts = async () => {

  setLoading(true);

  try {

    const list = [];

    await firebase.firestore()

      .collection('posts')

      .orderBy('postTime', 'desc')

      .get()

    .then((querySnapshot) => {

      // console.log('Total Posts: ', querySnapshot.size);

      querySnapshot.forEach((doc) => {

        const {

          userName,
          userId,
          post,
          postImg,
```

```
postTime,  
likes,  
comments,  
profileimg,
```

```
} = doc.data();
```

```
console.log("Poster Boy: ",postImg)
```

```
list.push({  
    id: doc.id,  
    userId,  
    userName: userName,  
    userImg:profileimg,  
    postTime: postTime,  
    post,  
    // postImg:'https://lh5.googleusercontent.com/-
```

```
b0PKyNuQv5s/AAAAAAAAAAI/AAAAAAAAM/AMZuuclxAM4M1SCBGAO7Rp-  
QP6zgBEUkOQ/s96-c/photo.jpg',
```

```
    postImg:postImg,  
    liked: false,  
    likes,  
    comments,  
  );  
});  
});  
  
setPosts(list);  
 setLoading(false);  
  
console.log('Posts: ', posts);  
} catch (e) {  
  console.log(e);  
}  
};  
  
useEffect(() => {  
  fetchPosts();  
}, []);
```

```
useEffect(() => {
```

```
    fetchPosts();
```

```
    setDeleted(false);
```

```
}, [deleted]);
```

```
const handleDelete = (postId) => {
```

```
    Alert.alert(
```

```
        'Delete post',
```

```
        'Are you sure?',
```

```
[
```

```
{
```

```
    text: 'Cancel',
```

```
    onPress: () => console.log('Cancel Pressed!'),
```

```
    style: 'cancel',
```

```
        },  
        {  
          text: 'Confirm',  
          onPress: () => deletePost(postId),  
        },  
      ],  
      {cancelable: false},  
    );  
  };
```

```
const deletePost = (postId) => {  
  console.log('Current Post Id: ', postId);  
  
  firebase.firestore()  
    .collection('posts')  
    .doc(postId)
```

```
.get()

.then((documentSnapshot) => {

  if (documentSnapshot.exists) {

    const {postImg} = documentSnapshot.data();

    if (postImg != null) {

      const storageRef = firebase.storage().refFromURL(postImg);

      const imageRef = firebase.storage().ref(storageRef.fullPath);

      imageRef

        .delete()

        .then(() => {

          console.log(`#${postImg} has been deleted successfully.`);

          deleteFirestoreData(postId);

          setDeleted(true);

        })

        .catch((e) => {

          console.log('Error while deleting the image. ', e);

        });

    }

  }

  // If the post image is not available
```

```
    } else {  
  
      deleteFirestoreData(postId);  
  
    }  
  
  }  
  
});  
  
};
```

```
const deleteFirestoreData = (postId) => {  
  
  firebase.firestore()  
  
    .collection('posts')  
  
    .doc(postId)  
  
    .delete()  
  
  .then(() => {  
  
    Alert.alert(  
      'Post deleted!',
```



```
<Button  
    title="Post a Job"  
    // onPress={() => navigation.navigate('Addpostscreen')}  
  
    onPress={() => {  
        navigation.navigate('Addpostscreen', {  
            fetchPosts, // Pass the function as a prop  
        });  
    }}  
/>  
  
<FlatList  
    data={posts}  
    renderItem={renderPostItem}  
    keyExtractor={(item) => item.id}  
    showsVerticalScrollIndicator={false}  
/>  
</Container>  
})
```

```
</SafeAreaView>

);

};

const styles = StyleSheet.create({
  card: {
    backgroundColor: '#e2ddd9',
    width: 350,
    marginBottom: 20,
    borderRadius: 10,
    padding: 15,
    height: 500,
  },
  userInfo: {
    flexDirection: 'row',
    justifyContent: 'flex-start',
    padding: 15,
  },
  userImg: {
```

```
width: 50,  
  
height: 50,  
  
borderRadius: 25,  
  
},  
  
userInfoText: {  
  
flexDirection: 'column',  
  
justifyContent: 'center',  
  
marginLeft: 10,  
  
},  
  
userName: {  
  
fontSize: 14,  
  
fontWeight: 'bold',  
  
},  
  
postTime: {  
  
fontSize: 12,  
  
color: '#666',  
  
},  
  
postText: {  
  
fontSize: 14,  
  
paddingLeft: 15,  
  
paddingRight: 15,
```

```
marginBottom: 15,  
},  
  
postImg: {  
  
width: '100%',  
  
height: 250,  
  
},  
  
divider: {  
  
borderBottomColor: '#dddddd',  
  
borderBottomWidth: 1,  
  
width: '92%',  
  
alignSelf: 'center',  
  
marginTop: 15,  
  
},  
  
interactionWrapper: {  
  
flexDirection: 'row',  
  
justifyContent: 'space-around',  
  
padding: 15,  
  
},  
  
interaction: {  
  
flexDirection: 'row',  
  
justifyContent: 'center',
```

```
borderRadius: 5,  
  
padding: 2,  
  
backgroundColor: 'transparent',  
  
},  
  
interactionText: {  
  
fontSize: 12,  
  
fontWeight: 'bold',  
  
marginTop: 5,  
  
marginLeft: 5,  
  
},  
  
activeInteraction: {  
  
backgroundColor: '#2e64e515',  
  
},  
  
activeInteractionText: {  
  
color: '#2e64e5',  
  
},  
  
});  
  
export default JobPost;
```

Jobs3.js:

```
// Jobs2.js

import React, { useState, useEffect } from "react";

import {
    Text,
    StyleSheet,
    Image,
    View,
    Pressable,
    FlatList,
    StatusBar,
    ActivityIndicator,
    TextInput,
    Button,
} from "react-native";

import { useNavigation } from "@react-navigation/native";

// import useFetch from "../hooks/useFetch";

import axios from 'axios';

import Spacing from "../constants1/Spacing";
```

```
import FontSize from "../constants1/FontSize";
import Colors from "../constants1/Colors";

const Jobs3 = () => {
  const navigation = useNavigation();

  const [searchTerm, setSearchTerm] = useState("");
  const [jobs, setJobs] = useState([]);
  const [isLoading, setIsLoading] = useState(false);

  const spacing = 20;
  const avatarSize = 70;

  const searchJobs = async () => {
    setLoading(true);
    try {
      const response = await axios.get('https://jsearch.p.rapidapi.com/search', {
        params: {
          query: searchTerm,
          // ...other API parameters as needed
        }
      });
      if (response.data.error) {
        setError(response.data.error);
      } else {
        setJobs(response.data.jobs);
      }
    } catch (error) {
      console.error(error);
    }
  };
}
```

```
headers: {  
    'X-RapidAPI-Key': '0e1cc89325mshbc3b1ea4b0b83b3p1d8c89jsn6e94b700e9ae',  
    'X-RapidAPI-Host': 'jsearch.p.rapidapi.com'  
}  
});  
  
if (response.data && response.data.data) {  
    setJobs(response.data.data); // Extract job listings  
} else {  
    console.warn('Invalid data format from API');  
}  
console.log(response.data); // Assuming the API returns data in this format  
} catch (error) {  
    console.error('Job search error:', error);  
    // Handle API errors appropriately  
} finally {  
    setIsLoading(false);  
}  
};
```

```
//  useEffect(() => {  
//    setJobs(null);  
//  }, []);  
  
try {  
  return (  
    <View style={{ backgroundColor: "#808080" }}>  
      {/* TextInput for searching */}  
  
      <TextInput  
        placeholder="Search for jobs"  
        value={searchTerm}  
        onChangeText={setSearchTerm}  
        onSubmitEditing={searchJobs}  
        style={[  
          //fontFamily: Font["poppins-regular"],  
          fontSize: FontSize.small,  
          padding: Spacing * 2,  
        ]}  
    </View>  
  );  
}  
catch (error) {  
  console.error(error);  
}
```

```
    backgroundColor: Colors.lightPrimary,  
  
    borderRadius: Spacing,  
  
    marginVertical: Spacing,  
  
    borderColor: 'black',  
  
    borderWidth: 1,  
  
},  
  
]  
}
```

/>

{

```
isLoading ? (  
  <ActivityIndicator size="large" color="blue" />
```

):(

<FlatList

data={jobs}

keyExtractor={(item) => item.index }

```
contentContainerStyle={{

    padding: spacing,

    paddingTop: StatusBar.currentHeight || 42,


}}



renderItem={({ item }) => {

    return (

        <Pressable

            onPress={() =>

                navigation.navigate("Job_des", {

                    job_name: item.job_title,

                    company: item.employer_name,

                    salary: item.job_city,

                    image: item.employer_logo,

                    applybefore: item.job_offer_expiration_datetime_utc,

                    jobtype: item.job_employment_type,

                    posted_on: item.job_posted_at_datetime_utc,

                    job_description: item.job_description,

                    job_qualityscore: item.job_apply_quality_score,

                    job_applylink: item.job_apply_link,

                    longitiude: item.job_longitude,

                    latitude: item.job_latitude,


                })
            }
        </Pressable>
    );
}}
```

```
        website: item.employer_website,  
    })  
  
}  
  
>  
  
<View  
  
    style={{  
        flexDirection: "row",  
        padding: spacing,  
        marginBottom: spacing,  
        borderRadius: 12,  
        backgroundColor: "rgba(255,255,255,0.8)",  
        shadowColor: "rgb(100,100,100)",  
        shadowOffset: {  
            width: 0,  
            height: 10,  
        },  
        shadowOpacity: 1,  
        shadowRadius: 20,  
    }}  
  
>  
  
<Image
```

```
source={ { uri: item.employer_logo } }

style={{

width: avatarSize,

height: avatarSize,

borderRadius: avatarSize,

marginRight: spacing / 2,




} }

/>

<View style={ { flex: 1 } }>

<Text

style={{

fontSize: 22,





fontWeight: "700",





} }

>

{item.job_title}

</Text>

<Text style={ { fontSize: 18, opacity: 0.7 } }>

{item.employer_name}

</Text>

<Text
```

```
        style={{fontSize: 14, opacity: 0.8, color: "#0099cc" }}
```

```
>
```

```
{item.job_city}
```

```
</Text>
```

```
</View>
```

```
</View>
```

```
</Pressable>
```

```
);
```

```
}}
```

```
/>
```

```
)
```

```
}
```

```
/* FlatList to display job data */
```

```
</View>
```

```
);
```

```
    } catch (error) {  
  
      console.error("Error in rendering Jobs2 component:", error);  
  
    }  
  
};
```

```
const styles = StyleSheet.create({
```

loadingContainer:

flex: 1,

justifyContent: "center",

alignItems: "center",

},

errorContainer: {

flex: 1,

justifyContent: "center",

alignItems: "center",

},

errorText: {

fontSize: 18,

`fontWeight: "bold",`

color: "red",

},

```
});  
  
export default Jobs3;
```

5.14Login.js:

```
import {  
  
  SafeAreaView,  
  
  StyleSheet,  
  
  Text,  
  
  TextInput,  
  
  TouchableOpacity,  
  
  View,  
  
} from "react-native";  
  
import React,{useState} from "react";  
  
import { AntDesign } from '@expo/vector-icons';  
  
import Spacing from "../constants1/Spacing";  
  
import FontSize from "../constants1/FontSize";  
  
import Colors from "../constants1/Colors";  
  
//import Font from "../constants/Font";
```

```
import { Ionicons } from "@expo/vector-icons";
import { NativeStackScreenProps } from "@react-navigation/native-stack";
import AppTextInput from "../components/AppTextInput";
import { ActivityIndicator } from "react-native";
import { signInWithEmailAndPassword } from "firebase/auth";
import { firebase } from "../config/firebase";
import { useNavigation } from "@react-navigation/native";

const Login = ({navigation}) => {
    //const navigation = useNavigation();
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const [focused, setFocused] = useState(false);
    const [loading, setLoading] = useState(false);
```

```
loginUser = async (email,password) => {
    try{
        setLoading(true);
        await firebase.auth().signInWithEmailAndPassword(email,password);
    } catch(err){
        alert(err.message);
    }finally {
        setLoading(false);
    }
}

return (
<SafeAreaView>
<View
style={{{
    padding: Spacing * 2,
}}}>
<View
style={{{
    alignItems: "center",
}}>
```

```
    } }

>

<Text

  style={{

    fontSize: FontSize.xLarge,

    color: Colors.primary,

    //fontFamily: Font["poppins-bold"],

    marginVertical: Spacing * 3,

  }}

>

  Login here

</Text>

<TouchableOpacity //133

  style={{

    padding: Spacing,

    backgroundColor: Colors.gray,

    borderRadius: Spacing / 2,

    marginHorizontal: Spacing,
```

```
        }

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

    >

    <AntDesign name="arrowleft" size={24} color="black" />

</TouchableOpacity>

<Text

    style={{

        //fontFamily: Font["poppins-semiBold"],

        fontSize: FontSize.large,

        maxWidth: "60%",

        textAlign: "center",

    }}>

    >

    Welcome back you've been missed!

</Text>

</View>

<View

    style={{

        marginVertical: Spacing * 3,

    }}>

    >
```

```
 {/* <AppTextInput placeholder="Email" onClick1={handleChangeemail} />

<AppTextInput placeholder="Password" onClick1={handleChangeepass} /> */}

<TextInput

onFocus={() => setFocused(true)}

onBlur={() => setFocused(false)}

placeholderTextColor={Colors.darkText}

placeholder= "Email"

onChangeText={(email) => setEmail(email)}

//onChangeText={(text) => setEmail(text)}

style={[

{

//fontFamily: Font["poppins-regular"],

fontSize: FontSize.small,

padding: Spacing * 2,

backgroundColor: Colors.lightPrimary,

borderRadius: Spacing,

marginVertical: Spacing,

borderColor: 'black',

borderWidth: 1,
```

```
 },  
 focused && {  
   borderWidth: 3,  
   borderColor: Colors.primary,  
   shadowOffset: { width: 4, height: Spacing },  
   shadowColor: Colors.primary,  
   shadowOpacity: 0.2,  
   shadowRadius: Spacing,  
,  
 }]  
 />
```

```
<TextInput  
  onFocus={() => setFocused(true)}  
  onBlur={() => setFocused(false)}  
  placeholderTextColor={Colors.darkText}  
  placeholder="Password"  
  value={password}
```

```
onChangeText={(password) => setPassword(password)}
```

```
//onChangeText={(text) => setPassword(text)}
```

```
secureTextEntry={true}
```

```
style={[
```

```
{
```

```
//fontFamily: Font["poppins-regular"],
```

```
fontSize: FontSize.small,
```

```
padding: Spacing * 2,
```

```
backgroundColor: Colors.lightPrimary,
```

```
borderRadius: Spacing,
```

```
marginVertical: Spacing,
```

```
borderColor: 'black',
```

```
borderWidth: 1,
```

```
},
```

```
focused && {
```

```
borderWidth: 3,
```

```
borderColor: Colors.primary,
```

```
shadowOffset: { width: 4, height: Spacing },
```

```
shadowColor: Colors.primary,
```

```
shadowOpacity: 0.2,
```

```
        shadowRadius: Spacing,  
    },  
]  
/  

```

```
</View>
```

```
<View>
```

```
    <Text  
        style={ {  
            //fontFamily: Font["poppins-semiBold"],  
            fontSize: FontSize.small,  
            color: Colors.primary,  
            alignSelf: "flex-end",  
        } }  
    >
```

Forgot your password ?

```
</Text>
```

```
</View>

{

    loading ? (

        <ActivityIndicator size="large" color="blue" />

    ):(
        <TouchableOpacity

            style={ {

                padding: Spacing * 2,

                backgroundColor: Colors.primary,

                marginVertical: Spacing * 3,

                borderRadius: Spacing,

                shadowColor: Colors.primary,

                shadowOffset: {

                    width: 0,

                    height: Spacing,

                },

                shadowOpacity: 0.3,

                shadowRadius: Spacing,

```

```
        } }

onPress={() => loginUser(email,password) }

>

<Text

style={{

// fontFamily: Font["poppins-bold"],

color: Colors.onPrimary,

textAlign: "center",

fontSize: FontSize.large,


} }

>

Log in

</Text>

</TouchableOpacity>

})
```

```
<TouchableOpacity  
    //onPress={() => navigate("Register")}  
  
    style={{  
        padding: Spacing,  
    }}  
  
>  
  
<Text  
    style={{  
        //fontFamily: Font["poppins-semiBold"],  
        color: Colors.text,  
        textAlign: "center",  
        fontSize: FontSize.small,  
    }}  
  
>  
  
Create new account  
  
</Text>  
  
</TouchableOpacity>
```

```
<View  
    style={{  
        marginVertical: Spacing * 3,
```

```
    } }

>

<Text

style={{

// fontFamily: Font["poppins-semiBold"],

color: Colors.primary,

textAlign: "center",

fontSize: FontSize.small,


}}
```

>

Or continue with

```
</Text>
```

```
<View
```

```
style={{

marginTop: Spacing,

flexDirection: "row",

justifyContent: "center",


}}
```

>

```
<TouchableOpacity
```

```
style={{

padding: Spacing,

backgroundColor: Colors.gray,

borderRadius: Spacing / 2,

marginHorizontal: Spacing,


} }

>

<Ionicons

name="logo-google"

color={Colors.text}

size={Spacing * 2}

/>

</TouchableOpacity>

<TouchableOpacity

style={{

padding: Spacing,

backgroundColor: Colors.gray,

borderRadius: Spacing / 2,

marginHorizontal: Spacing,


} }

>
```

```
<Ionicons  
    name="logo-apple"  
    color={Colors.text}  
    size={Spacing * 2}  
/>  
  
</TouchableOpacity>  
  
<TouchableOpacity  
    style={{  
        padding: Spacing,  
        backgroundColor: Colors.gray,  
        borderRadius: Spacing / 2,  
        marginHorizontal: Spacing,  
    }}  
>>  
  
<Ionicons  
    name="logo-facebook"  
    color={Colors.text}  
    size={Spacing * 2}  
/>  
  
</TouchableOpacity>  
  
</View>
```

```
</View>

</View>

</SafeAreaView>

);

};

export default Login;
```

5.15 Profile.js:

```
import { StyleSheet, Text, View, ImageBackground } from "react-native";

import Top from "../components/Top";

import Middle from "../components/Middle";

//import Bottom from "../components/Bottom";

import Sizes from "../constants/Sizes";



export default function Profile() {

  return (
```

```
</>

<ImageBackground

    style={styles.backgroundImage}

    source={require("../assets/bg.png")}

    // blurRadius={1}

>

<View style={styles.container}>

    <Top />

    <Middle />

    {/* <Bottom/> */}

</View>

</ImageBackground>

</>

);

}
```

```
const styles = StyleSheet.create({

    backgroundImage: {

        flex: 1,

        resizeMode: "cover",

    },
})
```

```
container: {  
  marginHorizontal: Sizes.medium,  
  marginTop: Sizes.safe,  
},  
});
```

5.16 Top.js:

```
import { StyleSheet, Text, View, TouchableOpacity } from "react-native";  
  
import React from "react";  
  
import { AntDesign } from "@expo/vector-icons";  
  
import Colors1 from "../constants/colors1";  
  
export default function Top() {  
  return (  
    <View style={styles.icons}>  
      <TouchableOpacity style={styles.setting}>  
        <AntDesign name="setting" size={35} color="white" />  
      </TouchableOpacity>  
    </View>  
  );  
}
```

```
</TouchableOpacity>

</View>

);

}

const styles = StyleSheet.create({

  icons: {

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

  },

  back:{

    backgroundColor: Colors1.alt,

    width: 45,

    height: 45,

    borderTopLeftRadius: 10,

    borderBottomRightRadius: 10,

    justifyContent: "center",

    alignItems: "center",

  }

})
```

```
});
```

5.17 Middle.js:

```
import { StyleSheet, Text, View, Image, Button, ActivityIndicator, SafeAreaView } from
"react-native";

import React, { useEffect, useState } from "react";

import { AntDesign } from "@expo/vector-icons";

import Colors1 from "../constants/colors1";

import { firebase } from "../config/firebase";

export default function Middle() {

  const [name, setName] = useState("");

  const [imageuri, setImageuri] = useState();

  const [imageKey, setImageKey] = useState(0);

  const [loading, setLoading] = useState(false);

  useEffect(() =>
```

```
{  
    firebase.firestore().collection('users')  
        .doc(firebase.auth().currentUser.uid)  
        .get()  
        .then((documentSnapshot) => {  
            if (documentSnapshot.exists) {  
                setName(documentSnapshot.data().name);  
                setImageuri(documentSnapshot.data().imageuri);  
                setImageKey(imageKey + 1);  
            }  
        }  
        else{  
            console.log('User does not exist');  
        }  
    })  
},[]);
```

```
const handleLogout = async () => {

  try {

    setLoading(true); // Set loading to true when logout operation starts

    await firebase.auth().signOut();

  } catch (error) {

    console.error('Error logging out:', error);

  } finally {

    setLoading(false); // Set loading to false when logout operation finishes

  }

};

return (

<View style={styles.main}>

<View style={styles.imageContainer}>

<Image style={styles.image} source= {{uri: imageuri}} key={imageKey}/>

<Text style={{ fontSize: 16, color: "white", fontWeight: "bold" }}>

  {name}

</Text>

<Text

  style={{ fontSize: 16, color: Colors1.darkGray, fontWeight: "500" }}>

>
```

```
    {/* {auth.currentUser?.email} */}

    {firebase.auth().currentUser?.email}

    </Text>

    </View>

{loading ? ( // Render loading indicator while logout operation is in progress

    <ActivityIndicator size="large" color="blue" />

) : (

<View style={styles.middleSectionTextContainer}>

    <Button title="Logout" color={'red'}
```

```
}

const styles = StyleSheet.create({
  main: {
    marginTop: 30,
  },
  imageContainer: {
    justifyContent: "center",
    alignItems: "center",
  },
  image: {
    width: 150,
    height: 150,
    borderRadius: 50,
    marginBottom: 5,
  },
  middleSectionTextContainer: {
    flexDirection: "row",
    justifyContent: "space-evenly",
    marginTop: 20,
  },
})
```

```
middleSectionText: {  
    justifyContent: "center",  
    alignItems: "center",  
},  
  
toptext: {  
    fontSize: 16,  
    color: Colors1.white,  
    fontWeight: "bold",  
},  
  
bottomtext: {  
    fontSize: 16,  
    color: Colors1.darkGray,  
    fontWeight: "700",  
},  
  
});
```

5.18 ResumeForm.js:

```
import * as React from "react";
```

```
import {  
  Text,  
  View,  
  StyleSheet,  
  TextInput,  
  Button,  
  TouchableOpacity,  
  ScrollView  
} from "react-native";  
  
import { useState } from "react";  
  
const ResumeForm = ({navigation}) => {  
  
  const [userDetails, setUserDetails] = useState({  
  
    fullName: "",  
  
    avatarUrl: "",  
  
    profTitle: "",  
  
    phoneNo: "",  
  
    email: "",  
  
    website: "",
```

```
company: "",  
jobTitle: "",  
jobStartDate: "",  
jobEndDate: "",  
experience: "",  
profSummary: "",  
certificate: "",  
collegeName: "",  
colStartDate: "",  
colEndDate: "",  
skill: "",  
hobby: "",  
});  
  
return (  
    <ScrollView>  
        <View style={styles.container}>  
            <View style={styles.header}>  
                <Text style={styles.headerText}>Resume Builder</Text>  
            </View>
```

```
<View style={styles.details}>

  <Text style={styles.titleText}>Personal Details</Text>

  <TextInput
    style={styles.textinput}
    placeholder="Enter your full name"
    value={userDetails.fullName}
    onChangeText={(e) => {
      setUserDetails((userDetails) => ({
        ...userDetails,
        ...{ fullName: e },
      }));
    }}
  />

  {/* <TextInput
    style={styles.textinput}
    placeholder="Enter your avatar URL"
    value={userDetails.avatarUrl} }}
```

```
onChangeText={(e) => {  
    setUserDetails((userDetails) => ({  
        ...userDetails,  
        ...{ avatarUrl: e },  
    }));  
}}  
> */}  
  
<TextInput  
    style={styles.textinput}  
    placeholder="Enter your professional title"  
    value={userDetails.profTitle}  
    onChangeText={(e) => {  
        setUserDetails((userDetails) => ({  
            ...userDetails,  
            ...{ profTitle: e },  
        }));  
}}  
>  
</View>
```

```
{/* <View style={styles.details}>

<Text style={styles.titleText}>Personal Details</Text>

<TextInput
    style={styles.textinput}
    placeholder="Enter your email address"
    value={userDetails.fullName}
    onChangeText={(e) => {
        setUserDetails((userDetails) => ({
            ...userDetails,
            ...{ fullName: e },
        }));
    }}
/>

<TextInput
    style={styles.textinput}
    placeholder="Enter your phone number"
```

```
        value={userDetails.avatarUrl}

        onChangeText={(e) => {

            setUserDetails((userDetails) => ({
                ...userDetails,
                ...{ avatarUrl: e },
            }));
        }}

    >

<TextInput

    style={styles.textinput}

    placeholder="Enter your address"

    value={userDetails.profTitle}

    onChangeText={(e) => {

        setUserDetails((userDetails) => ({
            ...userDetails,
            ...{ profTitle: e },
        }));
    }}

>

</View>

*/}
```

```
<View style={styles.details}>

    <Text style={styles.titleText}>Previous Job</Text>

    <TextInput
        style={styles.textinput}
        placeholder="Enter company name"
        value={userDetails.company}
        onChangeText={(e) => {
            setUserDetails((userDetails) => ({
                ...userDetails,
                ...{ company: e },
            }));
        }}
    />

    <TextInput
        style={styles.textinput}
        placeholder="Enter job title"
        value={userDetails.jobTitle}
        onChangeText={(e) => {
            setUserDetails((userDetails) => ({
```

```
...userDetails,  
...{ jobTitle: e },  
});  
}  
/>  
  
<TextInput  
style={styles.textinput}  
placeholder="Enter start date (e.g. 11/11/2022)"  
value={userDetails.jobStartDate}  
onChangeText={(e) => {  
    setUserDetails((userDetails) => ({  
        ...userDetails,  
        ...{ jobStartDate: e },  
    }));  
}}  
/>  
  
<TextInput  
style={styles.textinput}  
placeholder="Enter end date (e.g. 12/11/2022)"  
value={userDetails.jobEndDate}  
onChangeText={(e) => {
```

```
        setUserDetails((userDetails) => ({
          ...userDetails,
          ...{ jobEndDate: e },
        }));
      }

    />

<TextInput
  style={styles.textinput}
  placeholder="Describe your experience"
  value={userDetails.experience}
  onChangeText={(e) => {
    setUserDetails((userDetails) => ({
      ...userDetails,
      ...{ experience: e },
    }));
  }}
>
```

</View>

```
<Button  
    title="Create Resume"  
    style={styles.button}  
    onPress={() => navigation.navigate('ShowCV', userDetails)}  
>  
</Button>  
  
</View>  
  
</ScrollView>  
);  
};  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#36485f',  
    paddingLeft: 40,  
    paddingRight: 40,  
  },  
});
```

```
paddingTop: 40  
},  
  
header: {  
  
marginBottom: 20,  
  
alignSelf: 'stretch'  
},  
  
details: {  
  
marginBottom: 15  
},  
  
headerText: {  
  
fontSize: 24,  
  
color: '#fff',  
  
borderBottomColor: '#199187',  
  
paddingBottom: 10,  
  
borderBottomWidth: 1  
},  
  
titleText: {  
  
fontWeight: 'bold',  
  
color: 'yellow',  
  
fontSize: 15,  
  
marginBottom: 10
```

```
    },  
  
    textinput: {  
        alignSelf: 'stretch',  
        height: 40,  
        color: '#4DF717',  
        marginBottom: 20,  
        borderBottomColor: '#f8f8f8',  
        borderBottomWidth: 1  
    },  
  
    button: {  
        alignSelf: 'stretch',  
        alignItems: 'center',  
        padding: 10,  
        backgroundColor: '#4DF717',  
        marginTop: 5,  
        marginBottom: 20,  
    }  
});  
  
export default ResumeForm;
```

5.19 ShowCv.js:

```
import React from 'react';

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

import * as Print from 'expo-print';

import * as FileSystem from 'expo-file-system';

export default function ShowCV({ route }) {

  let dataObj = route.params;

  const handlePrintToFile = async () => {

    try {

      const htmlContent = generateHTMLContent(dataObj);

      const pdf = await Print.printToFileAsync({ html: htmlContent });

      if (Platform.OS === 'ios') {

        FileSystem.getContentUriAsync(pdf.uri).then((uri) => {

          Print.printAsync({ uri: uri.uri });

        });

      }

    }

  };

}


```

```
});  
}  
  
    Print.printAsync({ uri: pdf.uri });  
  
}  
  
} catch (error) {  
  
    console.error('Error printing:', error);  
  
}  
  
};
```

```
const generateHTMLContent = (data) => {  
  
    return `  
  
        <html>  
  
            <head>  
  
                <title>Your Resume</title>  
  
                <style>  
  
                    ${stylesToString(styles)}  
  
                </style>  
  
            </head>  
  
            <body>  
  
                <div class="container">  
  
                    <div class="header">
```

```
<h1>Your Resume</h1>

</div>

<div class="details">

<h2>Personal Details</h2>



<p><strong>Name:</strong> ${data.fullName}</p>

<p><strong>Professional Title:</strong> ${data.profTitle}</p>

</div>

</div>

</body>

</html>

`;

};

const stylesToString = (styles) => {

  let styleStr = "";

  for (const key in styles) {

    if (styles.hasOwnProperty(key)) {

      const styleObj = styles[key];

      styleStr += `${key} {`;

      for (const styleKey in styleObj) {

        styleStr += `${styleKey}: ${styleObj[styleKey]};`;
      }
    }
  }
}
```

```
        if (styleObj.hasOwnProperty(styleKey)) {  
  
            styleStr += `${styleKey}:${styleObj[styleKey]};`;  
  
        }  
  
    }  
  
    styleStr += '}';  
  
}  
  
}  
  
return styleStr;  
  
};  
  
  
  
return (  
  
    <View style={styles.container}>  
  
        <View style={styles.header}>  
  
            <Text style={styles.headerText}>Your Resume</Text>  
  
        </View>  
  
  
  
        <View style={styles.details}>  
  
            <Text style={styles.titleText}>Personal Details</Text>  
  
            <Image source={{ uri: dataObj.avatarUrl }} style={{ width: 80, height: 80 }} />  
  
            <Text style={styles.text}>  
  
                <Text style={styles.key}>Name: </Text>
```

```
<Text>{dataObj.fullName}</Text>

</Text>

<Text style={styles.text}>

<Text style={styles.key}>Professional Title: </Text>

<Text>{dataObj.profTitle}</Text>

</Text>

</View>

<Button title="Print to File" onPress={handlePrintToFile} />

</View>

);

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#36485f',

    paddingLeft: 40,

    paddingRight: 40,

    paddingTop: 40,
```

```
},  
  
header: {  
  
  marginBottom: 20,  
  
  alignSelf: 'stretch',  
  
},  
  
details: {  
  
  marginBottom: 15,  
  
},  
  
headerText: {  
  
  fontSize: 24,  
  
  color: '#fff',  
  
  borderBottomColor: '#199187',  
  
  paddingBottom: 10,  
  
  borderBottomWidth: 1,  
  
},  
  
titleText: {  
  
  fontWeight: 'bold',  
  
  color: 'yellow',  
  
  fontSize: 15,  
  
  marginBottom: 10,  
  
},
```

```
key: {  
  fontWeight: 'bold',  
},  
  
text: {  
  color: '#d3d3d3',  
},  
});
```

5.20 SignUp.js:

```
import React from 'react';  
  
import { Text, View, StyleSheet, Image, Button, Platform } from 'react-native';  
  
import * as Print from 'expo-print';  
  
import * as FileSystem from 'expo-file-system';  
  
  
  
  
export default function ShowCV({ route }) {  
  
  let dataObj = route.params;  
  
  
  
  
  const handlePrintToFile = async () => {  
  
    try {
```

```
const htmlContent = generateHTMLContent(dataObj);

const pdf = await Print.printToFileAsync({ html: htmlContent });

if (Platform.OS === 'ios') {

  FileSystem.getContentUriAsync(pdf.uri).then((uri) => {

    Print.printAsync({ uri: uri.uri });

  });

} else {

  Print.printAsync({ uri: pdf.uri });

}

} catch (error) {

  console.error('Error printing:', error);

}

};

const generateHTMLContent = (data) => {

  return `

<html>

<head>

<title>Your Resume</title>

<style>
```

```
    ${stylesToString(styles)}
```

```
  </style>
```

```
  </head>
```

```
  <body>
```

```
    <div class="container">
```

```
      <div class="header">
```

```
        <h1>Your Resume</h1>
```

```
      </div>
```

```
      <div class="details">
```

```
        <h2>Personal Details</h2>
```

```
        
```

```
        <p><strong>Name:</strong> ${data.fullName}</p>
```

```
        <p><strong>Professional Title:</strong> ${data.profTitle}</p>
```

```
      </div>
```

```
    </div>
```

```
  </body>
```

```
</html>
```

```
 `;
```

```
};
```

```
const stylesToString = (styles) => {
```

```
let styleStr = "";

for (const key in styles) {

  if (styles.hasOwnProperty(key)) {

    const styleObj = styles[key];

    styleStr += `${key}{`;

    for (const styleKey in styleObj) {

      if (styleObj.hasOwnProperty(styleKey)) {

        styleStr += `${styleKey}:${styleObj[styleKey]};`;

      }

    }

    styleStr += '}';

  }

}

return styleStr;

};

return (

<View style={styles.container}>

<View style={styles.header}>

<Text style={styles.headerText}>Your Resume</Text>

</View>
```

```
<View style={styles.details}>

    <Text style={styles.titleText}>Personal Details</Text>

    <Image source={{ uri: dataObj.avatarUrl }} style={{ width: 80, height: 80 }} />

    <Text style={styles.text}>

        <Text style={styles.key}>Name: </Text>

        <Text>{dataObj.fullName}</Text>

    </Text>

    <Text style={styles.text}>

        <Text style={styles.key}>Professional Title: </Text>

        <Text>{dataObj.profTitle}</Text>

    </Text>

</View>

<Button title="Print to File" onPress={handlePrintToFile} />

</View>

);

}

const styles = StyleSheet.create({
```

```
container: {  
  flex: 1,  
  
  backgroundColor: '#36485f',  
  
  paddingLeft: 40,  
  
  paddingRight: 40,  
  
  paddingTop: 40,  
  
},  
  
header: {  
  
  marginBottom: 20,  
  
  alignSelf: 'stretch',  
  
},  
  
details: {  
  
  marginBottom: 15,  
  
},  
  
headerText: {  
  
  fontSize: 24,  
  
  color: '#fff',  
  
  borderBottomColor: '#199187',  
  
  paddingBottom: 10,  
  
  borderBottomWidth: 1,  
  
},
```

```
        titleText: {  
          fontWeight: 'bold',  
          color: 'yellow',  
          fontSize: 15,  
          marginBottom: 10,  
        },  
  
        key: {  
          fontWeight: 'bold',  
        },  
  
        text: {  
          color: '#d3d3d3',  
        },  
      });
```

VideoProfile.js:

```
import * as React from "react";  
  
import { Image, StyleSheet, Text, View, Pressable, SafeAreaView, ScrollView,
```

```
ActivityIndicator,Button } from "react-native";\n\nimport { Card } from "@ui-kitten/components";\n\nimport { ViewPropTypes } from 'deprecated-react-native-prop-types';\n\n\n\n\nimport { firebase } from './config.firebaseio';\n\nimport { useEffect, useState } from "react";\n\nimport { Video, ResizeMode } from 'expo-av';\n\nimport * as FileSystem from 'expo-file-system';\n\nconst VideoProfile = () => {\n\n  const [videos, setVideos] = useState(null);\n\n  const video = React.useRef(null);\n\n  const [status, setStatus] = React.useState({});\n\n  // const [loading, setLoading] = useState(true); // New loading state\n\n\n\n\nuseEffect(() => {\n\n  const fetchVideoURI = async () => {\n\n    try{\n\n      const snapshot = await\n\n      firebase.firestore().collection('users').doc(firebase.auth().currentUser.uid).get();\n\n    }\n\n  }\n\n};
```

```
if(snapshot.exists){

    const data = snapshot.data();

    setVideos(data.video);

}

else{

    console.log('User does not exist');

}

}

catch (error) {

    console.error('Error fetching video URI:', error);

}

};

fetchVideoURI();

}, []);
```

```
const handleDownload = async () => {

  if (!videos) {

    console.log('No video to download.');

    return;
  }

  try {

    const downloadResumable = FileSystem.createDownloadResumable(
      videos,
      FileSystem.documentDirectory + 'downloaded_video.mp4'
    );

    const { uri } = await downloadResumable.downloadAsync();

    console.log('Video downloaded to:', uri);

    // You can now use the downloaded video URI as needed, such as displaying it in your
    app.

  } catch (error) {

    console.error('Error downloading video:', error);

  }
};
```

```
return (

<View style={styles.container}>

<Video

ref={video}

style={styles.video}

source={{

// uri: 'https://d23dyxeqlo5psv.cloudfront.net/big_buck_bunny.mp4',

uri: videos,

} }

useNativeControls

resizeMode={ResizeMode.CONTAIN}

isLooping

onPlaybackStatusUpdate={status => setStatus(() => status)}

/>

<View style={styles.buttons}>

<Button

title={status.isPlaying ? 'Pause' : 'Play'}

onPress={() =>

status.isPlaying ? video.current.pauseAsync() : video.current.playAsync()

}

}
```

```
/>

</View>

<Button
    title= 'Download'
    onPress={handleDownload}
/>

</View>

);

};

export default VideoProfile;

const styles = StyleSheet.create({
    container: {
        flex: 1,
        alignItems: 'center',
        justifyContent: 'center',
    }
});
```

```
    },  
  
    video: {  
  
        flex:1,  
  
        alignSelf: 'stretch',  
  
    },  
  
    buttons: {  
  
        marginBottom: 40,  
  
        marginTop: 15,  
  
    },  
  
});
```

VideoRecord.js:

```
import React from "react";  
  
import { useState, useEffect } from "react";  
  
import { StyleSheet, Text, View, Button, Pressable } from "react-native";  
  
import { Camera } from "expo-camera";  
  
import { Video } from "expo-av";
```

```
import {firebase} from "../config.firebaseio";

export default function VideoRecord() {

  const [hasAudioPermission, setHasAudioPermission] = useState(null);

  const [hasCameraPermission, setHasCameraPermission] = useState(null);

  const [camera, setCamera] = useState(null);

  const [record, setRecord] = useState(null);

  const [type, setType] = useState(Camera.Constants.Type.back);

  const video = React.useRef(null);

  const [status, setStatus] = React.useState({ });

  const [rm, setRm] = useState(true);

  useEffect(() => {

    (async () => {

      const cameraStatus = await Camera.requestCameraPermissionsAsync();

      setHasCameraPermission(cameraStatus.status === "granted");

      const audioStatus = await Camera.requestMicrophonePermissionsAsync();

      setHasAudioPermission(audioStatus.status === "granted");

    })();

  }, []);

  const takeVideo = async () => {
```



```
if(currentUserId)

{

    const useref = firebase.firestore().collection("users").doc(currentUserId);

    await useref.update({

        video: data.uri

    });

    console.log("Video saved to Firestore");

}

// const videosCollection = firebase.firestore().collection("videos");



// await videosCollection.add({

//     uri: data.uri,

//     createdAt: firebase.firestore.FieldValue.serverTimestamp()

// });

// console.log("Video saved to Firestore");
```

```
        } catch (error) {

            console.error("Error saving video to Firestore: ", error);

        }

    };

const stopVideo = async () => {

    camera.stopRecording();

    setRm(true);

};

if (hasCameraPermission === null || hasAudioPermission === null) {

    return <View />;

}

if (hasCameraPermission === false || hasAudioPermission === false) {

    return <Text>No access to camera</Text>;

}

return (

<View style={{ flex: 1 }}>

<View style={styles.cameraContainer}>
```

```
<Camera  
    ref={ref => setCamera(ref)}  
    style={styles.fixedRatio}  
    type={type}  
    ratio={"6:3"}  
/>  
</View>  
  
{ rm ?(<Video  
    ref={video}  
    style={styles.video}  
    source={{  
        uri: record,  
    }}  
    useNativeControls  
    resizeMode="contain"  
    isLooping  
    onPlaybackStatusUpdate={status => setStatus(() => status)}  
/> ) : null }  
  
<View style={styles.buttons}>
```

```
<Button  
    title={status.isPlaying ? "Pause" : "Play"}  
  
    onPress={() =>  
  
        status.isPlaying  
  
        ? video.current.pauseAsync()  
  
        : video.current.playAsync()  
  
    }  
  
/>  
  
</View>  
  
  
  
  
<Pressable style={styles.button}  
  
    onPress={() => {  
  
        setType(  
  
            type === Camera.Constants.Type.back  
  
            ? Camera.Constants.Type.front  
  
            : Camera.Constants.Type.back  
  
        );  
  
    }}>  
  
<Text style={styles.text}>Flip Video</Text>  
  
</Pressable>
```

```
<Pressable style={styles.button} onPress={() => takeVideo()}>  
  <Text style={styles.text}>Take Video</Text>  
</Pressable>  
  
<Pressable style={styles.button} onPress={() => stopVideo()}>  
  <Text style={styles.text}>Stop Video</Text>  
</Pressable>  
  
<Pressable style={styles.button} onPress={() => setRm(!rm)}>  
  <Text style={styles.text}>Cross Video</Text>  
</Pressable>  
  
</View>  
);  
}
```

```
const styles = StyleSheet.create({
```

```
  cameraContainer: {
```

```
    flex: 1,
```

```
    flexDirection: "row",
```

```
},  
  
fixedRatio: {  
  
    flex: 1,  
  
    aspectRatio: 1,  
  
},  
  
video: {  
  
    alignSelf: "center",  
  
    width: 350,  
  
    height: 220,  
  
},  
  
buttons: {  
  
    flexDirection: "row",  
  
    justifyContent: "center",  
  
    alignItems: "center",  
  
},  
  
button: {  
  
    alignItems: 'center',  
  
    justifyContent: 'center',  
  
    paddingVertical: 12,
```

```
paddingHorizontal: 32,  
borderRadius: 4,  
elevation: 3,  
backgroundColor: 'black',  
},  
text: {  
fontSize: 16,  
lineHeight: 21,  
fontWeight: 'bold',  
letterSpacing: 0.25,  
color: 'white',  
},  
});
```

Welcome.js:

```
import {View, Text, Button} from 'react-native';  
  
import React from 'react';  
  
import { StyleSheet } from 'react-native';
```

```
import { Image } from 'react-native';

import { LinearGradient } from 'expo-linear-gradient';

//import { COLORS } from './constants/colors';

const Welcome = ({navigation}) => {

  const handle = () => {

    console.log('Button pressed');

  }

  return (
    <LinearGradient style = {{

      flex: 1,

    }}>

      colors={['#1E90FF', '#FFA500']}

    </LinearGradient>
  );
}
```

```
<View style={styles.container}>

<View style={styles.circleview}>
  <View style={styles.circleimg}>
    <Image source={require('../assets/1.png')}>
      style={{

        width: 200, height: 200, borderRadius: 200/2,
      }}/>
    </View>
  </View>
</View>

<View>
  <Button color="#ff5c5c" title='Welcome' style = {{

    //marginBottom: 20,
    position: 'absolute',
    left: 100,
  }}>
    Welcome
  </Button>
</View>
```

```
        top: 500,  
        width: 50,  
    } }  
  
    onPress={() => navigation.navigate('Signup')}  
  
    //onPress={() => navigation.navigate('Jobs2')}  
  
    />  
  
<Button title='Explore Jobs' style = {{  
    //marginBottom: 20,  
    width: "50%",  
} }  
  
    //onPress={() => navigation.navigate('Signup')}  
  
    onPress={handle}  
  
    />  
  
</View>  
  
</LinearGradient>  
)
```

```
    }
```

```
const styles = StyleSheet.create({
```

```
  container: {
```

```
    flex: 1,
```

```
    alignItems: 'center',
```

```
    justifyContent: 'center',
```

```
  },
```

```
  circleview: {
```

```
    width: 250,
```

```
    height: 250,
```

```
    borderRadius: 250/2,
```

```
    backgroundColor: "#FFFFFF",
```

```
    position: 'absolute',
```

```
    // left: -120,
```

```
    // top: -20,
```

```
  },
```

```
circleimg :{

  display: 'flex',

  alignItems: 'center',

  justifyContent: 'center',

}

});

export default Welcome;
```

AppTextInput.js:

```
import {

  StyleSheet,
  Text,
  TextInput,
  View,
} from "react-native";

//import {Signup} from '../screens';

import React, { useState } from "react";
```

```
import Colors from "../constants1/Colors";
import Font from "../constants1/Font";
import FontSize from "../constants1/FontSize";
import Spacing from "../constants1/Spacing";

const AppTextInput = (props) =>{
  const [focused, setFocused] = useState(false);

  return (
    <TextInput
      onFocus={() => setFocused(true)}
      onBlur={() => setFocused(false)}
      placeholderTextColor={Colors.darkText}
      placeholder= {props.placeholder}
      onChange={props.onClick1}
      style={[
        {
          //fontFamily: Font["poppins-regular"],
          fontSize: FontSize.small,
          padding: Spacing * 2,
          backgroundColor: Colors.lightPrimary,
        }
      ]}
    />
  );
}
```

```
marginVertical: Spacing,  
borderColor: 'black',  
borderWidth: 1,  
,  
focused && {  
borderWidth: 3,  
borderColor: Colors.primary,  
shadowOffset: { width: 4, height: Spacing },  
shadowColor: Colors.primary,  
shadowOpacity: 0.2,  
shadowRadius: Spacing,  
,  
]  
/  
);  
};  
  
export default AppTextInput;
```

```
const styles = StyleSheet.create({});
```

PostCard.js:

```
import React, { useContext, useEffect, useState } from "react";  
  
import Ionicons from "react-native-vector-icons";  
  
import {  
    Card,  
    Container,  
    Divider,  
    Interaction,  
    InteractionText,  
    InteractionWrapper,  
    PostText,  
    UserImg,  
    UserInfo,  
    UserInfoText,  
    UserName,  
    PostTime,  
    PostImg,  
}
```

```
    } from "../styles/FeedStyles";  
  
export default function PostCard({item}) {  
  
  likeIcon = item.liked ? 'heart' : 'heart-outline';  
  
  likeIconColor = item.liked ? 'red' : 'black';  
  
  if (item.likes == 1) {  
    likeText = '1 Like';  
  
  } else if (item.likes > 1) {  
    likeText = item.likes + ' Likes';  
  
  } else {  
    likeText = 'Like';  
  
  }  
  
  if (item.comments == 1) {  
    commentText = '1 Comment';  
  } else {  
    commentText = item.comments + ' Comments';  
  }  
}  
}
```

```
        } else if (item.comments > 1) {  
  
            commentText = item.comments + ' Comments';  
  
        } else {  
  
            commentText = 'Comment';  
  
        }  
  
    }  
  
    return (  
        <Card>  
  
        <UserInfo>  
  
            <UserImg source={item.userImg} />  
  
            <UserInfoText>  
  
                <UserName>{item.userName}</UserName>  
  
                <PostTime>{item.postTime}</PostTime>  
  
            </UserInfoText>  
  
        </UserInfo>  
  
        <PostText>{item.post}</PostText>  
    );  
}
```

```
{item.postImg != "none" ? (  
    <PostImg source={item.postImg} />  
    ) : (  
        <Divider />  
    )}  
  
/* <PostImg source={require("../assets/posts/post-img-2.jpg")}> */  
  
<Divider />  
  
<InteractionWrapper>  
    <Interaction active={item.liked}>  
        <Ionicons name={likeIcon} size={25} color={likeIconColor} />  
        <InteractionText active={item.liked}>{likeText}</InteractionText>  
    </Interaction>  
  
<Interaction>  
    <Ionicons name="md-chatbubble-outline" size={25} color="red" />  
    <InteractionText>commentText</InteractionText>  
    </Interaction>  
</InteractionWrapper>
```

```
</Card>

);

}

}
```

Firebase.js:

```
// Import the functions you need from the SDKs you need

// import { initializeApp } from "firebase/app";

// import { getAuth } from "firebase/auth";

//import { getFirestore } from "firebase/firestore";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

import firebase from 'firebase/compat/app';

import 'firebase/compat/auth';

import 'firebase/compat/firestore';

import 'firebase/compat/storage';

// Your web app's Firebase configuration

const firebaseConfig = {
```

```
apiKey: "AIzaSyCzWVodPrvGCc_Aiy25tYMD-lPHWjN-GbI",  
authDomain: "authentication-92f91.firebaseio.com",  
projectId: "authentication-92f91",  
storageBucket: "authentication-92f91.appspot.com",  
messagingSenderId: "442027400703",  
appId: "1:442027400703:web:a81ecdf20abd738d861bca"  
};  
  
if (!firebase.apps.length) {  
  firebase.initializeApp(firebaseConfig);  
}  
  
export { firebase};
```

Colors.js:

```
const COLORS = {  
  primary: "#1E90FF",
```

```
secondary: "#FFA500",  
  
white: "#FFFFFF",  
  
black: "#000000",  
  
gray: "#808080"  
}
```

```
export default COLORS;
```

```
const Colors1 = {  
  
light: "#eeee",  
  
grey: "#D8D8D8",  
  
darkGray: "#3F4F57",  
  
primary: "#99D5F3",  
  
secondary: "#F7BE7D",  
  
dark: "#000",  
  
white: "#ffff",  
  
alt: "rgba(0,0,0,0.5)",  
};
```

```
export default Colors1;
```

AddPost.js:

```
import styled from 'styled-components';

export const InputWrapper = styled.View`  
  flex: 1;  
  
  justify-content: center;  
  
  align-items: center;  
  
  width: 100%;  
  
  background-color: #2e64e515;  
  
`;
```

```
export const InputField = styled.TextInput`  
  
  justify-content: center;  
  
  align-items: center;  
  
  font-size: 24px;  
  
  text-align: center;  
  
  width: 90%;  
  
  margin-bottom: 15px;  
  
`;
```

```
export const AddImage = styled.Image`  
  width: 100%;  
  
  height: 250px;  
  
  margin-bottom: 15px;  
`;
```

```
export const StatusWrapper = styled.View`  
  justify-content: center;  
  
  align-items: center;  
`;
```

```
export const SubmitBtn = styled.TouchableOpacity`  
  flex-direction: row;  
  
  justify-content: center;  
  
  background-color: #2e64e515;  
  
  border-radius: 5px;  
  
  padding: 10px 25px;  
`;
```

```
export const SubmitBtnText = styled.Text`
```

```
font-size: 18px;  
${" /* font-family: 'Lato-Bold'; */}  
  
font-weight: bold;  
  
color: #2e64e5;  
  
`;
```

FeedStyles.js:

```
import styled from 'styled-components';
```

```
export const Container = styled.View`
```

```
flex: 1;  
  
align-items: center;  
  
background-color: #fff;  
  
padding: 20px;  
  
`;
```

```
export const Card = styled.View`
```

```
background-color: #f8f8f8;
```

```
width: 100%;  
  
margin-bottom: 20px;  
  
border-radius: 10px;  
  
padding: 25px;  
  
`;
```

```
export const UserInfo = styled.View`  
  
flex-direction: row;  
  
justify-content: flex-start;  
  
padding: 15px;  
  
`;
```

```
export const UserImg = styled.Image`  
  
width: 50px;  
  
height: 50px;  
  
border-radius: 25px;  
  
`;
```

```
export const UserInfoText = styled.View`  
  
flex-direction: column;  
  
justify-content: center;
```

```
margin-left: 10px;  
`;  
  
export const UserName = styled.Text`  
  font-size: 14px;  
  font-weight: bold;  
  
`;
```

```
export const PostTime = styled.Text`  
  font-size: 12px;  
  color: #666;  
`;
```

```
export const PostText = styled.Text`  
  font-size: 14px;  
  padding-left: 15px;  
  padding-right: 15px;  
  margin-bottom: 15px;  
`;
```

```
export const PostImg = styled.Image`  
  width: 100%;  
  
  height: 250px;  
  
  /* margin-top: 15px; */  
  
`;
```

```
export const Divider = styled.View`  
  
  border-bottom-color: #dddddd;  
  
  border-bottom-width: 1px;  
  
  width: 92%;  
  
  align-self: center;  
  
  margin-top: 15px;  
  
`;
```

```
export const InteractionWrapper = styled.View`  
  
  flex-direction: row;  
  
  justify-content: space-around;  
  
  padding: 15px;  
  
`;
```

```
export const Interaction = styled.TouchableOpacity`
```

```
flex-direction: row;  
justify-content: center;  
border-radius: 5px;  
padding: 2px 5px;  
background-color: ${props => props.active ? '#2e64e515' : 'transparent'}  
;  
  
export const InteractionText = styled.Text`  
font-size: 12px;  
  
font-weight: bold;  
color: ${props => props.active ? '#2e64e5' : '#333'};  
margin-top: 5px;  
margin-left: 5px;  
`;
```

GlobalStyles.js:

```
/* fonts */
```

```
// export const FontFamily = {  
//   interBold: "Inter-Bold",  
//   interLight: "Inter-Light",  
//   robotoRegular: "Roboto-Regular",  
//   aPPM10: "Montserrat-Medium",  
//   aPPTTextButton14SB: "Montserrat-SemiBold",  
//   aPPTTextButton14R: "Montserrat-Regular",  
//   sFSubheadlineSemibold: "SF Pro Text",  
//   poppinsSemibold: "Poppins-SemiBold",  
//   poppinsRegular: "Poppins-Regular",  
//   poppinsMedium: "Poppins-Medium",  
//   poppinsBold: "Poppins-Bold",  
//   futuraMdBT: "Futura Md BT",  
//   futuraHvBT: "Futura Hv BT",  
//   robotoMedium: "Roboto-Medium",  
// };  
  
/* font sizes */  
  
export const FontSize = {  
  sFSubheadlineSemibold_size: 15,  
  aPPLabel12M_size: 12,  
  size_11xl: 30,
```

```
aPPM10_size: 10,  
aPPTitle520M_size: 20,  
aPP14Medium_size: 14,  
aPPIInput16R_size: 16,  
size_smi: 13,  
size_xs_5: 12,  
size_7xl: 26,  
size_9xl: 28,  
size_lg: 18,  
size_21xl: 40,  
};  
/* Colors */  
export const Color = {  
  colorSteelblue_100: "#5c93b3",  
  colorSteelblue_200: "#488aa6",  
  primaryBLACK: "#000",  
  primaryWHITE: "#fff",  
  colorGainsboro_100: "#e3e3e3",  
  colorGainsboro_200: "#d9d9d9",  
  colorGainsboro_300: "rgba(217, 217, 217, 0.27)",  
  colorDarkslategray_100: "#053c40",
```

```
shadesOfBlackGrey2: "#666",  
  
shadesOfBlackGrey3: "#828282",  
  
colorGray_600: "#1e1e1e",  
  
textColor: "#181a1f",  
  
colorGray_200: "rgba(0, 0, 0, 0.5)",  
  
colorGray_400: "rgba(0, 0, 0, 0.1)",  
  
colorGray_100: "rgba(0, 0, 0, 0.3)",  
  
colorGray_300: "rgba(0, 0, 0, 0.05)",  
  
colorGray_500: "rgba(0, 0, 0, 0.49)",  
  
successErrorBLUE: "#1477f8",  
  
e1: "#eaf0ff",  
  
shadesOfBlackGrey5: "#d2d2d2",  
  
colorLightblue: "#abd0e5",  
  
colorGoldenrod: "#daa400",  
  
colorSeagreen: "#07864b",  
  
bG: "#f5f7fc",  
  
colorGhostwhite_100: "#f0f3fc",  
  
primary: "#1c58f2",  
  
colorDarkgray: "#c1adad",  
  
};  
  
/* Paddings */
```

```
export const Padding = {
```

```
  p_9xs: 4,
```

```
  p_5xs: 8,
```

```
  p_7xs: 6,
```

```
  p_5xl: 24,
```

```
  p_11xl: 30,
```

```
  p_xl: 20,
```

```
  p_base: 16,
```

```
  p_xs: 12,
```

```
  p_10xs: 3,
```

```
  p_11xs: 2,
```

```
};
```

```
/* border radiiuses */
```

```
export const Border = {
```

```
  br_11xl: 30,
```

```
  br_31xl: 50,
```

```
  br_81xl: 100,
```

```
  br_xs: 12,
```

```
  br_sm: 14,
```

```
  br_11xs: 2,
```

```
  br_9xs: 4,
```

```
    br_6xl: 25,  
  
    br_lgi: 19,  
  
    br_114xl: 133,  
  
    br_mid: 17,  
  
    br_4xl: 23,  
  
    br_7xs: 6,  
  
    br_base: 16,  
  
    br_5xl: 24,  
  
    br_xl: 20,  
  
    br_6xs: 7,  
  
    br_72xl_5: 92,  
  
};
```

Package.json:

```
{  
  
  "name": "proconnect",  
  
  "version": "1.0.0",  
  
  "main": "node_modules/expo/AppEntry.js",  
  
  "scripts": {  
  
    "start": "expo start -c",
```

```
"android": "expo start --android",  
  
"ios": "expo start --ios",  
  
"web": "expo start --web"  
  
},  
  
"dependencies": {  
  
"@expo/config": "^8.1.1",  
  
"@expo/metro-config": "^0.10.0",  
  
"@expo/vector-icons": "^13.0.0",  
  
"@firebase/storage": "^0.12.3",  
  
"@gluestack-ui/themed": "^1.1.17",  
  
"@react-native-async-storage/async-storage": "1.18.2",  
  
"@react-native-community/masked-view": "^0.1.11",  
  
"@react-native-firebase/app": "^18.9.0",  
  
"@react-native-firebase/auth": "^18.9.0",  
  
"@react-native-firebase/firestore": "^18.8.0",  
  
"@react-native-firebase/storage": "^18.9.0",  
  
"@react-native-masked-view/masked-view": "0.2.9",  
  
"@react-native-voice/voice": "^3.2.4",  
  
"@react-navigation/drawer": "^6.6.6",  
  
"@react-navigation/native": "^6.1.9",  
  
"@react-navigation/native-stack": "^6.9.17",
```

```
"@react-navigation/stack": "^6.3.20",  
"@ui-kitten/components": "^5.3.1",  
"axios": "^1.6.8",  
"deprecated-react-native-prop-types": "^5.0.0",  
"expo": "^49.0.23",  
"expo-av": "~13.4.1",  
"expo-camera": "~13.4.4",  
"expo-dev-client": "~2.4.13",  
"expo-file-system": "~15.4.5",  
"expo-image": "~1.3.5",  
"expo-image-picker": "~14.3.2",  
"expo-intent-launcher": "~10.7.0",  
"expo-linear-gradient": "~12.3.0",  
"expo-location": "~16.1.0",  
"expo-mail-composer": "~12.3.0",  
"expo-permissions": "~14.2.1",  
"expo-print": "~12.4.2",  
"expo-sharing": "^11.5.0",  
"expo-status-bar": "~1.6.0",  
"firebase": "^10.10.0",  
"lodash": "^4.17.21",
```

```
"moment": "^2.30.1",  
"react": "18.2.0",  
"react-native": "0.72.10",  
"react-native-action-button": "^2.8.5",  
"react-native-document-picker": "^9.1.1",  
"react-native-expo-image-cache": "^4.1.0",  
"react-native-gesture-handler": "~2.12.0",  
"react-native-image-crop-picker": "^0.40.3",  
"react-native-reanimated": "~3.3.0",  
"react-native-safe-area-context": "4.6.3",  
"react-native-screens": "~3.22.0",  
"react-native-share": "^10.1.0",  
"react-native-skeleton-placeholder": "^5.2.4",  
"react-native-vector-icons": "^10.0.3",  
"react-native-video": "^2.3.1",  
"react-native-video-player": "^0.14.0",  
"react-native-webview": "^13.2.2",  
"react-navigation": "^5.0.0",  
"short-url": "^0.0.1",  
"styled-components": "^6.1.8"  
},
```

```

"devDependencies": {

  "@babel/core": "^7.20.0"

  },


  "private": true


}

log("DOMParser not supported, falling back to createHTMLDocument");doc =
document.implementation.createHTMLDocument(""); try { doc.open();
doc.write(html); doc.close(); } catch
(ee) {

  log("createHTMLDocument write not supported, falling back to
document.body.innerHTML");

  doc.body.innerHTML = html; // ie9 doesnt support writing to
documentElement

}

var b = doc.querySelector("base"); if (!b ||
!b.href.host) { var base =
doc.createElement("base"); base.href = src;
doc.head.insertBefore(base, doc.head.firstChild);

}

return doc;
};

}

function labelCanvasElements(ownerDocument) {

```

```
        [].slice.call(ownerDocument.querySelectorAll("canvas"), 0).forEach(function  
        (canvas) {            canvas.setAttribute(html2canvasCanvasCloneAttribute,  
        "canvas-" + html2canvasCanvasCloneIndex++);  
    });  
}  
  
function cloneCanvasContents(ownerDocument, documentClone) {  
    [].slice.call(ownerDocument.querySelectorAll("[ " +  
html2canvasCanvasCloneAttribute + "]"), 0).forEach(function (canvas) {  
    try {
```

```

        var clonedCanvas = documentClone.querySelector('[' +
html2canvasCanvasCloneAttribute + '="" +
canvas.getAttribute(html2canvasCanvasCloneAttribute) + "]');

        if (clonedCanvas) {
clonedCanvas.width = canvas.width;
clonedCanvas.height = canvas.height;

clonedCanvas.getContext("2d").putImageData(canvas.getContext("2d").getImageDat
a(0, 0, canvas.width, canvas.height), 0, 0);
    }

} catch (e) {      log("Unable to copy canvas
content from", canvas, e);
}

canvas.removeAttribute(html2canvasCanvasCloneAttribute);
});

}

function removeScriptNodes(parent) {
[]).slice.call(parent.childNodes, 0).filter(isElementNode).forEach(function (node)
{
    if (node.tagName === "SCRIPT")
{
    parent.removeChild(node);
}
else {
removeScriptNodes(node);
}
});

return parent;
}

function isElementNode(node) {      return
node.nodeType === Node.ELEMENT_NODE;
}

```

```

        function absoluteUrl(url) {      var link =
document.createElement("a");      link.href
= url;      link.href = link.href;
return link;
}

// http://dev.w3.org/csswg/css-color/

function Color(value) {      this.r =
0;      this.g = 0;      this.b = 0;
this.a = null;      var result =
this.fromArray(value) ||
this.namedColor(value) ||
this.rgb(value) ||      this.rgba(value)
||      this.hex6(value) ||
this.hex3(value);
}

Color.prototype.darken = function (amount) {
var a = 1 - amount;      return new Color([
Math.round(this.r * a),
Math.round(this.g * a),
Math.round(this.b * a),
this.a
]);
};

Color.prototype.isTransparent = function () {
return this.a === 0;
};

Color.prototype.isBlack = function () { return this.r ===
0 && this.g === 0 && this.b === 0;
};

```

```
};

Color.prototype.fromArray = function (array) {
if (Array.isArray(array)) {      this.r =
Math.min(array[0], 255);      this.g =
Math.min(array[1], 255);      this.b =
Math.min(array[2], 255);      if (array.length
> 3) {      this.a = array[3];
}
}
```

```
return (Array.isArray(array));
};
```

```
var _hex3 = /^[a-f0-9]{3}$/i;
```

```
Color.prototype.hex3 = function (value) {      var
match = null;      if ((match = value.match(_hex3))
!== null) {      this.r = parseInt(match[1][0] +
match[1][0], 16);      this.g = parseInt(match[1][1]
+ match[1][1], 16);      this.b =
parseInt(match[1][2] + match[1][2], 16);
}
return match !== null;
};
```

```
var _hex6 = /^[a-f0-9]{6}$/i;
```

```
Color.prototype.hex6 = function (value) {      var
match = null;      if ((match = value.match(_hex6))
!== null) {      this.r =
parseInt(match[1].substring(0, 2), 16);      this.g =
parseInt(match[1].substring(2, 4), 16);
this.b = parseInt(match[1].substring(4, 6), 16);
}
353
```

```

    }
    return match !== null;
};

var _rgb = /^rgb\(((d{1,3}) *, *(d{1,3}) *, *(d{1,3}))$/;

Color.prototype.rgb = function (value) {
var match = null;      if ((match =
value.match(_rgb)) !== null) {      this.r =
Number(match[1]);      this.g =
Number(match[2]);      this.b =
Number(match[3]);
}
return match !== null;
};

var _rgba = /^rgba\(((d{1,3}) , *(d{1,3}) *, *(d{1,3}) *, *(d+.?\d))$/;

Color.prototype.rgba = function (value) {
var match = null;      if ((match =
value.match(_rgba)) !== null) {      this.r =
Number(match[1]);      this.g =
Number(match[2]);      this.b =
Number(match[3]);      this.a =
Number(match[4]);
}
return match !== null;
};

Color.prototype.toString = function () { return
this.a !== null && this.a !== 1 ?
"rgba(" + [this.r, this.g, this.b, this.a].join(",") + ")" :
"rgb(" + [this.r, this.g, this.b].join(",") + ")";
};

```

```
};
```

```
Color.prototype.namedColor = function (value) {
var color = colors[value.toLowerCase()];      if
(color) {          this.r = color[0];          this.g =
color[1];          this.b = color[2];
} else if (value.toLowerCase() === "transparent") {
this.r = this.g = this.b = this.a = 0;      return true;
}

return !!color;
};

Color.prototype.isColor = true;

// JSON.stringify([]).slice.call($$('.named-color-table tr'), 1).map(function(row) {
return [row.childNodes[3].textContent,
row.childNodes[5].textContent.trim().split(",").map(Number)]
}).reduce(function(data, row) {data[row[0]] = row[1]; return data}, {}))    var
colors = {
  "aliceblue": [240, 248, 255],
  "antiquewhite": [250, 235, 215],
  "aqua": [0, 255, 255],
  "aquamarine": [127, 255, 212],
  "azure": [240, 255, 255],
  "beige": [245, 245, 220],
  "bisque": [255, 228, 196],
  "black": [0, 0, 0],
  "blanchedalmond": [255, 235, 205],
  "blue": [0, 0, 255],
  "blueviolet": [138, 43, 226],
  "brown": [165, 42, 42],
  "burlywood": [222, 184, 135],
  "cadetblue": [95, 158, 160],
```

"chartreuse": [127, 255, 0],
"chocolate": [210, 105, 30],
"coral": [255, 127, 80],
"cornflowerblue": [100, 149, 237],
"cornsilk": [255, 248, 220],
"crimson": [220, 20, 60],
"cyan": [0, 255, 255],
"darkblue": [0, 0, 139],
"darkcyan": [0, 139, 139],
"darkgoldenrod": [184, 134, 11],
"darkgray": [169, 169, 169],
"darkgreen": [0, 100, 0],
"darkgrey": [169, 169, 169],
"darkkhaki": [189, 183, 107],
"darkmagenta": [139, 0, 139],
"darkolivegreen": [85, 107, 47],
"darkorange": [255, 140, 0],
"darkorchid": [153, 50, 204],
"darkred": [139, 0, 0],
"darksalmon": [233, 150, 122],
"darkseagreen": [143, 188, 143],
"darkslateblue": [72, 61, 139],
"darkslategray": [47, 79, 79],
"darkslategrey": [47, 79, 79],
"darkturquoise": [0, 206, 209],
"darkviolet": [148, 0, 211],
"deeppink": [255, 20, 147],
"deepskyblue": [0, 191, 255],
"dimgray": [105, 105, 105],
"dimgrey": [105, 105, 105], "dodgerblue": [30, 144, 255],
"firebrick": [178, 34, 34],
"floralwhite": [255, 250, 240],
"forestgreen": [34, 139, 34],

"fuchsia": [255, 0, 255],
"gainsboro": [220, 220, 220],
"ghostwhite": [248, 248, 255],
"gold": [255, 215, 0],
"goldenrod": [218, 165, 32],
"gray": [128, 128, 128],
"green": [0, 128, 0],
"greenyellow": [173, 255, 47],
"grey": [128, 128, 128],
"honeydew": [240, 255, 240],
"hotpink": [255, 105, 180],
"indianred": [205, 92, 92],
"indigo": [75, 0, 130],
"ivory": [255, 255, 240],
"khaki": [240, 230, 140],
"lavender": [230, 230, 250],
"lavenderblush": [255, 240, 245],
"lawngreen": [124, 252, 0],
"lemonchiffon": [255, 250, 205],
"lightblue": [173, 216, 230],
"lightcoral": [240, 128, 128],
"lightcyan": [224, 255, 255],
"lightgoldenrodyellow": [250, 250, 210],
"lightgray": [211, 211, 211],
"lightgreen": [144, 238, 144],
"lightgrey": [211, 211, 211],
"lightpink": [255, 182, 193],
"lightsalmon": [255, 160, 122],
"lightseagreen": [32, 178, 170],
"lightskyblue": [135, 206, 250], "lightslategray": [119, 136, 153],
"lightslategrey": [119, 136, 153], "lightsteelblue":
[176, 196, 222],
"lightyellow": [255, 255, 224],

"lime": [0, 255, 0],
"limegreen": [50, 205, 50],
"linen": [250, 240, 230],
"magenta": [255, 0, 255],
"maroon": [128, 0, 0],
"mediumaquamarine": [102, 205, 170],
"mediumblue": [0, 0, 205],
"mediumorchid": [186, 85, 211],
"mediumpurple": [147, 112, 219],
"mediumseagreen": [60, 179, 113],
"mediumslateblue": [123, 104, 238],
"mediumspringgreen": [0, 250, 154],
"mediumturquoise": [72, 209, 204],
"mediumvioletred": [199, 21, 133],
"midnightblue": [25, 25, 112],
"mintcream": [245, 255, 250],
"mistyrose": [255, 228, 225],
"moccasin": [255, 228, 181],
"navajowhite": [255, 222, 173],
"navy": [0, 0, 128],
"oldlace": [253, 245, 230],
"olive": [128, 128, 0],
"olivedrab": [107, 142, 35],
"orange": [255, 165, 0],
"orangered": [255, 69, 0],
"orchid": [218, 112, 214],
"palegoldenrod": [238, 232, 170],
"palegreen": [152, 251, 152],
"paleturquoise": [175, 238, 238],
"palevioletred": [219, 112, 147], "papayawhip": [255, 239, 213],
"peachpuff": [255, 218, 185],
"peru": [205, 133, 63],
"pink": [255, 192, 203],

```
"plum": [221, 160, 221],  
"powderblue": [176, 224, 230],  
"purple": [128, 0, 128],  
"rebeccapurple": [102, 51, 153],  
"red": [255, 0, 0],  
"rosybrown": [188, 143, 143],  
"royalblue": [65, 105, 225],  
"saddlebrown": [139, 69, 19],  
"salmon": [250, 128, 114],  
"sandybrown": [244, 164, 96],  
"seagreen": [46, 139, 87],  
"seashell": [255, 245, 238],  
"sienna": [160, 82, 45],  
"silver": [192, 192, 192],  
"skyblue": [135, 206, 235],  
"slateblue": [106, 90, 205],  
    "slategray": [112, 128, 144],  
    "slategrey": [112, 128, 144],  
    "snow": [255, 250, 250],  
    "springgreen": [0, 255, 127],  
    "steelblue": [70, 130, 180],  
    "tan": [210, 180, 140],  
    "teal": [0, 128, 128],  
    "thistle": [216, 191, 216],  
    "tomato": [255, 99, 71],  
    "turquoise": [64, 224, 208],  
    "violet": [238, 130, 238],  
    "wheat": [245, 222, 179],  
    "white": [255, 255, 255],  
    "whitesmoke": [245, 245, 245], "yellow": [255, 255, 0],  
    "yellowgreen": [154, 205, 50]  
};
```

```

function DummyImageContainer(src) {
    this.src = src;
    log("DummyImageContainer for", src);
    if (!this.promise || !this.image) {
        log("Initiating DummyImageContainer");
        DummyImageContainer.prototype.image = new Image();
        var image = this.image;
        DummyImageContainer.prototype.promise = new Promise(function (resolve,
            reject) {
            image.onload = resolve;
            image.onerror = reject;
            image.src = smallImage();
            if (image.complete === true) {
                resolve(image);
            }
        });
    }
}

function Font(family, size) {
    var container = document.createElement('div'),
        img = document.createElement('img'),
        span
    = document.createElement('span'),
        sampleText = 'Hidden Text',
        baseline,
        middle;

    container.style.visibility = "hidden";
    container.style.fontFamily
    = family;
    container.style.fontSize = size;
    container.style.margin = 0;
    container.style.padding = 0;

    document.body.appendChild(container);

    img.src = smallImage();
    img.width = 1;
    img.height
    = 1;
}

```

```

    img.style.margin = 0;
    img.style.padding = 0;
    img.style.verticalAlign = "baseline";

    span.style.fontFamily = family;
    span.style.fontSize = size;
    span.style.margin = 0;      span.style.padding
    = 0;

    span.appendChild(document.createTextNode(sampleText));
    container.appendChild(span);      container.appendChild(img);
    baseline = (img.offsetTop - span.offsetTop) + 1;

    container.removeChild(span);
    container.appendChild(document.createTextNode(sampleText));

    container.style.lineHeight = "normal";
    img.style.verticalAlign = "super";

    middle = (img.offsetTop - container.offsetTop) + 1;

    document.body.removeChild(container);

    this.baseline = baseline; this.lineWidth
    = 1;
    this.middle = middle;
}

function FontMetrics() {
this.data = { };
}

```

```

FontMetrics.prototype.getMetrics = function (family, size) {
  if (this.data[family + "-" + size] === undefined) {
    this.data[family + "-" + size] = new Font(family, size);
  }
  return this.data[family + "-" + size];
};

function FrameContainer(container, sameOrigin, options) {
  this.image = null;
  this.src = container;
  var self = this;
  var bounds = getBounds(container);
  this.promise = (!sameOrigin ?
    this.proxyLoad(options.proxy, bounds, options) :
    new Promise(function (resolve) {
      if (container.contentWindow.document.URL === "about:blank" ||
        container.contentWindow.document.documentElement === null) {
        container.contentWindow.onload = container.onload = function () {
          resolve(container);
        };
      } else {
        resolve(container);
      }
    })).then(function (container) {
      return html2canvas(container.contentWindow.document.documentElement, {
        type: 'view',
        width: container.width,
        height: container.height,
        proxy: options.proxy,
        javascriptEnabled: options.javascriptEnabled,
        removeContainer: options.removeContainer,
        allowTaint: options.allowTaint,
        imageTimeout: options.imageTimeout / 2
      });
    }).then(function (canvas) {
      return self.image = canvas;
    });
}

```

```
}
```

```
FrameContainer.prototype.proxyLoad = function (proxy, bounds, options) {
var container = this.src;      return loadUrlDocument(container.src, proxy,
container.ownerDocument, bounds.width, bounds.height, options);
};
```

```
function GradientContainer(imageData) {
this.src = imageData.value;
this.colorStops = [];    this.type = null;
this.x0 = 0.5;      this.y0 = 0.5;
this.x1 = 0.5;      this.y1 = 0.5;
this.promise = Promise.resolve(true);
}
```

```
GradientContainer.prototype.TYPES = {
LINEAR: 1,
RADIAL: 2
};
```

```
function ImageContainer(src, cors) {    this.src = src;
this.image = new Image();    var self = this;
this.tainted = null;    this.promise = new
Promise(function (resolve, reject) {
self.image.onload = resolve;    self.image.onerror =
reject;    if (cors) {        self.image.crossOrigin =
"anonymous";
}    self.image.src = src;
if (self.image.complete === true) {
resolve(self.image);
}
});
```

```

        function ImageLoader(options, support) {
this.link = null;      this.options = options;
this.support = support;      this.origin =
this.getOrigin(window.location.href);
}

ImageLoader.prototype.findImages = function (nodes) {
var images = [];      nodes.reduce(function (imageNodes,
container) {      switch (container.nodeType) {
case "IMG":
return imageNodes.concat([{
args: [container.nodeType],
method: "url"
}]);
case "svg":      case
"IFRAME":
return imageNodes.concat([{
args: [container.nodeType],      method:
container.nodeType
}]);
}
return imageNodes;
}, []).forEach(this.addImage(images, this.loadImage), this);
return images;
};

ImageLoader.prototype.findBackgroundImage = function (images, container) {

container.parseBackgroundImages().filter(this.hasImageBackground).forEach(this.ad
dImage(images, this.loadImage), this);      return images;
};

ImageLoader.prototype.addImage = function (images, callback) {
return function (newImage) {

```

```

newImage.args.forEach(function (image) {           if
(!this.imageExists(images, image)) {
    images.splice(0, 0, callback.call(this, newImage));
log('Added image #' + (images.length), typeof(image) === "string" ?
image.substring(0, 100) : image);
}
}, this);
};

};

ImageLoader.prototype.hasImageBackground = function ( imageData ) {
return imageData.method !== "none";
};

ImageLoader.prototype.loadImage = function ( imageData ) {      if
(imageData.method === "url") {           var src = imageData.args[0];           if
(this.isSVG(src) && !this.support.svg && !this.options.allowTaint) {
return new SVGContainer(src);
} else if (src.match(/data:image\/*;base64,/i)) {           return new
ImageContainer(src.replace(/\url\([""]\{0,\}|[""]\{0,\}\$/ig, ")), false);
} else if (this.isSameOrigin(src) || this.options.allowTaint === true ||
this.isSVG(src)) {           return new ImageContainer(src, false);
} else if (this.support.cors && !this.options.allowTaint &&
this.options.useCORS) {           return new
ImageContainer(src, true);
} else if (this.options.proxy) {           return new
ProxyImageContainer(src, this.options.proxy);
} else {
    return new DummyImageContainer(src);
}
} else if (imageData.method === "linear-gradient") {
return new LinearGradientContainer(imageData);      }

```

```

else if (imageData.method === "gradient") {
    return new WebkitGradientContainer(imageData);
} else if (imageData.method === "svg") {
    return new SVGNodeContainer(imageData.args[0], this.support.svg);
} else if (imageData.method === "IFRAME") {
    return new FrameContainer(imageData.args[0],
        this.isSameOrigin(imageData.args[0].src), this.options);
} else {
    return new DummyImageContainer(imageData);
}
};


```

```

ImageLoader.prototype.isSVG = function (src) {
    return src.substring(src.length - 3).toLowerCase() === "svg" ||
    SVGContainer.prototype.isInline(src);
};


```

```

ImageLoader.prototype.imageExists = function (images, src) {
    return images.some(function (image) {
        return image.src
        === src;
    });
};


```

```

ImageLoader.prototype.isSameOrigin = function (url) {
    return (this.getOrigin(url) === this.origin);
};


```

```

ImageLoader.prototype.getOrigin = function (url) {
    var link =
    this.link || (this.link = document.createElement("a"));
    link.href = url;
    link.href = link.href; // IE9, LOL! - http://jsfiddle.net/niklasvh/2e48b/
    return link.protocol + link.hostname + link.port;
};


```

```

    ImageLoader.prototype.getPromise = function (container) {
        return
        this.timeout(container, this.options.imageTimeout)['catch'](function () {
            var
            dummy = new DummyImageContainer(container.src);
            return
            dummy.promise.then(function (image) {
                container.image = image;
            });
        });
    };

    ImageLoader.prototype.get = function (src) {
        var found = null;
        return
        this.images.some(function (img) {
            return (found = img).src === src;
        }) ? found : null;
    };

    ImageLoader.prototype.fetch = function (nodes) {
        this.images
        = nodes.reduce(bind(this.findBackgroundImage, this),
        this.findImages(nodes));
        this.images.forEach(function (image,
        index) {
            image.promise.then(function () {
                log("Successfully loaded image #" + (index + 1), image);
            }, function (e) {
                log("Failed loading image #"
                + (index + 1), image, e);
            });
        });
        this.ready = Promise.all(this.images.map(this.getPromise, this));
        log("Finished searching images");
        return this;
    };

    ImageLoader.prototype.timeout = function (container, timeout) {
        var
        timer;
        var promise = Promise.race([container.promise, new
        Promise(function (res, reject) {
            timer = setTimeout(function () {
                log("Timed out loading image", container);
                reject(container);
            }, timeout);
        })]);
    };

```

```

        }]).then(function (container) {
      clearTimeout(timer);      return
      container;
    });
    promise['catch'](function () {
      clearTimeout(timer);
    });
    return promise;
  };

  function LinearGradientContainer(imageData) {
    GradientContainer.apply(this, arguments);      this.type
    = this.TYPES.LINEAR;

    var hasDirection = imageData.args[0].match(this.stepRegExp) === null;

    if (hasDirection) {      imageData.args[0].split(
      ").reverse().forEach(function (position) {      switch (position) {
      case "left":*
        * jQuery FlexSlider v2.2.2
        * Copyright 2012 WooThemes
        * Contributing Author: Tyler Smith
        */
      ;
    (function ($) {

      //FlexSlider: Object Instance
      $.flexslider = function (el, options) {
      var slider = $(el);

      // making variables public      slider.vars =
      $.extend({ }, $.flexslider.defaults, options);

```

```

var namespace = slider.vars.namespace,
    msGesture = window.navigator && window.navigator.msPointerEnabled &&
window.MSGesture,          touch = (( "ontouchstart" in window ) || msGesture |||
window.DocumentTouch
&& document instanceof DocumentTouch) && slider.vars.touch,
    // deprecating this idea, as devices are being released with both of these events
//eventType = (touch) ? "touchend" : "click",           eventType = "click touchend
MSPointerUp",           watchedEvent = "",           watchedEventClearTimer,
vertical = slider.vars.direction === "vertical",       reverse = slider.vars.reverse,
carousel = (slider.vars.itemWidth > 0),           fade = slider.vars.animation === "fade",
asNav = slider.vars.asNavFor !== "",           methods = { },           focused = true;

    // Store a reference to the slider object
$.data(el, "flexslider", slider);

    // Private slider methods
methods = {
    init: function
() {
    slider.animating =
false;
    // Get current slide and make sure it is a number
slider.currentSlide = parseInt(( slider.vars.startAt ? slider.vars.startAt : 0),
10);           if (isNaN(slider.currentSlide)) slider.currentSlide
= 0;           slider.animatingTo = slider.currentSlide;
    slider.atEnd = (slider.currentSlide === 0 || slider.currentSlide ===
slider.last);           slider.containerSelector =
slider.vars.selector.substr(0, slider.vars.selector.search(' '));
    slider.slides = $(slider.vars.selector, slider);           slider.container =
$(slider.containerSelector, slider);           slider.count =
slider.slides.length;           // SYNC:
    slider.syncExists = $(slider.vars.sync).length > 0;
// SLIDE:

```

```

if (slider.vars.animation === "slide") slider.vars.animation = "swing";
slider.prop = (vertical) ? "top" : "marginLeft"; slider.args = { };
// SLIDESHOW:
slider.manualPause = false;
slider.stopped = false;
//PAUSE WHEN INVISIBLE
slider.started = false; slider.startTimeout = null; //
TOUCH/USECSS: slider.transitions = !slider.vars.video && !fade &&
slider.vars.useCSS &&
(function () {
var obj =
document.createElement('div'), props =
['perspectiveProperty', 'WebkitPerspective',
'MozPerspective', 'OPerspective', 'msPerspective'];
for (var i in props) if (obj.style[props[i]] !==
undefined) {
slider.pfx = props[i].replace('Perspective',
").toLowerCase(); slider.prop = "-" + slider.pfx + "-transform";
return true;
}
}())
return
false;
}());
// CONTROLSCONTAINER:
if (slider.vars.controlsContainer !== "") slider.controlsContainer =
$(slider.vars.controlsContainer).length > 0 && $(slider.vars.controlsContainer);
// MANUAL:
if (slider.vars.manualControls !== "") slider.manualControls =
$(slider.vars.manualControls).length > 0 && $(slider.vars.manualControls);

// RANDOMIZE:
if (slider.vars.randomize) {
slider.slides.sort(function () {
return
(Math.round(Math.random()) - 0.5);
});
}

```

```

        slider.container.empty().append(slider.slides);
    }

    slider.doMath();

    // INIT
    slider.setup("init");

    // CONTROLNAV:
    if (slider.vars.controlNav) methods.controlNav.setup();

    // DIRECTIONNAV:           if
    (slider.vars.directionNav) methods.directionNav.setup();

    // KEYBOARD:           if (slider.vars.keyboard &&
    ($slider.containerSelector).length === 1 || slider.vars.multipleKeyboard)) {
        $(document).bind('keyup', function (event) {           var
        keycode = event.keyCode;           if (!slider.animating && (keycode
        === 39 || keycode === 37)) {           var target = (keycode === 39)
        ? slider.getTarget('next') :           (keycode === 37) ?
        slider.getTarget('prev') : false;           slider.flexAnimate(target,
        slider.vars.pauseOnAction);

        }
    });
}

// MOUSEWHEEL:
if (slider.vars.mousewheel) {
    slider.bind('mousewheel', function (event, delta, deltaX, deltaY) {
        event.preventDefault();           var target = (delta < 0) ?
        slider.getTarget('next') :
        slider.getTarget('prev');

        slider.flexAnimate(target, slider.vars.pauseOnAction);

    });
}

```

```

// PAUSEPLAY      if (slider.vars.pausePlay)
methods.pausePlay.setup();

//PAUSE WHEN INVISIBLE      if
(slider.vars.slideshow && slider.vars.pauseInvisible)
methods.pauseInvisible.init();

// SLIDESHOW      if (slider.vars.slideshow) {
if (slider.vars.pauseOnHover) {           slider.hover(function () {
if (!slider.manualPlay && !slider.manualPause) slider.pause();
}, function () {           if (!slider.manualPause &&
!slider.manualPlay && !slider.stopped) slider.play();
});
}

// initialize animation
//If we're visible, or we don't use PageVisibility API      if
(!slider.vars.pauseInvisible || !methods.pauseInvisible.isHidden()) {
    (slider.vars.initDelay > 0) ? slider.startTimeout =
setTimeout(slider.play, slider.vars.initDelay) : slider.play();
}

// ASNAV:
if (asNav) methods.asNav.setup();

// TOUCH      if (touch &&
slider.vars.touch) methods.touch();      //
FADE&&SMOOTHHEIGHT || SLIDE:

if (!fade || (fade && slider.vars.smoothHeight)) $(window).bind("resize
orientationchange focus", methods.resize);

slider.find("img").attr("draggable", "false");

```

```

// API: start() Callback
setTimeout(function () {
  slider.vars.start(slider);
}, 200);
}, asNav: {
  setup: function () {
    slider.asNav = true;
    slider.animatingTo =
      Math.floor(slider.currentSlide / slider.move);
    slider.currentItem =
      slider.currentSlide;
    slider.slides.removeClass(namespace +
      "activeslide").eq(slider.currentItem).addClass(namespace + "active-slide");
    if (!msGesture) {
      slider.slides.on(eventType, function (e) {
        e.preventDefault();
        var $slide = $(this),
          target = $slide.index();
        var posFromLeft = $slide.offset().left - $(slider).scrollLeft(); // Find
        position of slide relative to left of slider container
        if (posFromLeft
          <= 0 && $slide.hasClass(namespace + 'activeslide')) {
          slider.flexAnimate(slider.getTarget("prev"), true);
        } else if (!$(slider.vars.asNavFor).data('flexslider').animating &&
          !$slide.hasClass(namespace + "active-slide")) {
          slider.direction
          = (slider.currentItem < target) ? "next" : "prev";
          slider.flexAnimate(target, slider.vars.pauseOnAction, false, true);
        }
      });
    } else {
      el._slider = slider;
      slider.slides.each(function () {
        var that = this;
        that._gesture = new MSGesture();
        that;
        that.addEventListener("MSPointerDown", function
        (e) {
          e.preventDefault();
          if
          (e.currentTarget._gesture)
            e.currentTarget._gesture.addPointer(e.pointerId);
        }, false);
        that.addEventListener("MSGestureTap", function (e) {
          e.preventDefault();
          var $slide = $(this),

```



```

        var captn = slide.attr('data-thumbcaption');

if (" != captn && undefined != captn) item += '<span class="' + namespace +
'caption">' + captn + '</span>';

    }

slider.controlNavScaffold.append('<li>' + item + '</li>');

j++;

    }

}

// CONTROLSCONTAINER:

(slider.controlsContainer) ?

$(slider.controlsContainer).append(slider.controlNavScaffold) :

slider.append(slider.controlNavScaffold);

methods.controlNav.set();

methods.controlNav.active();

slider.controlNavScaffold.delegate('a, img', eventType, function (event) {

event.preventDefault();

    if (watchedEvent === "" || watchedEvent === event.type) {

var $this = $(this), target =
slider.controlNav.index($this);

        if (!$this.hasClass(namespace + 'active')) {

slider.direction = (target > slider.currentSlide) ? "next" : "prev";
slider.flexAnimate(target, slider.vars.pauseOnAction);

    }

}

// setup flags to prevent event duplication

if (watchedEvent === "") {

watchedEvent = event.type;

}

```

```

methods.setToClearWatchedEvent();

});

},
setupManual: function () {
    slider.controlNav = slider.manualControls;
methods.controlNav.active();

    slider.controlNav.bind(eventType, function (event) {
event.preventDefault();

        if (watchedEvent === "" || watchedEvent === event.type) {
var $this = $(this), target =
slider.controlNav.index($this);

        if (!$this.hasClass(namespace + 'active')) {
            (target > slider.currentSlide) ? slider.direction = "next" :
slider.direction = "prev"; slider.flexAnimate(target,
slider.vars.pauseOnAction);

        }
    }

    // setup flags to prevent event duplication
if (watchedEvent === "") {
watchedEvent = event.type;
}

    methods.setToClearWatchedEvent();
});

}, set: function () {
var selector
= (slider.vars.controlNav === "thumbnails") ? 'img' : 'a';
slider.controlNav = $('. ' + namespace + 'control-nav li ' + selector,
(slider.controlsContainer) ? slider.controlsContainer : slider);
}, active: function () {
slider.controlNav.removeClass(namespace +

```

```

"active").eq(slider.animatingTo).addClass(namespace +
"active");
},
update: function (action, pos) {
  if (slider.pagingCount > 1 && action === "add") {
    slider.controlNavScaffold.append($('<li><a>' + slider.count +
      '</a></li>'));
  } else if (slider.pagingCount === 1) {
    slider.controlNavScaffold.find('li').remove();
  } else {
    slider.controlNav.eq(pos).closest('li').remove();
  }
  methods.controlNav.set();
  (slider.pagingCount > 1 && slider.pagingCount !==
  slider.controlNav.length) ? slider.update(pos, action) : methods.controlNav.active();
}
},
directionNav: { setup: function () { var
  directionNavScaffold = $('<ul class=' + namespace + 'directionnav"><li><a
  class=' + namespace + 'prev" href="#">' + slider.vars.prevText +
  '</a></li><li><a class=' + namespace + 'next" href="#">' + slider.vars.nextText
  + '</a></li></ul>');
}

// CONTROLSCONTAINER:
if (slider.controlsContainer) {
  $(slider.controlsContainer).append(directionNavScaffold);
  slider.directionNav = $('. ' + namespace + 'direction-nav li a',
  slider.controlsContainer);
} else { slider.append(directionNavScaffold);
  slider.directionNav = $('. ' + namespace + 'direction-nav li a', slider);
}

methods.directionNav.update();

```

```

slider.directionNav.bind(eventType, function (event) {
event.preventDefault();           var target;

    if (watchedEvent === "" || watchedEvent === event.type) {
target = ($(this).hasClass(namespace + 'next')) ?
slider.getTarget('next') : slider.getTarget('prev');

        slider.flexAnimate(target, slider.vars.pauseOnAction);

    }

// setup flags to prevent event duplication
    if (watchedEvent === "") {

watchedEvent = event.type;
    }

    methods.setToClearWatchedEvent();
});

},           update: function () {           var disabledClass =
namespace + 'disabled';           if (slider.pagingCount === 1) {
slider.directionNav.addClass(disabledClass).attr('tabindex', '-1');

} else if (!slider.vars.animationLoop) {           if
(slider.animatingTo === 0) {
slider.directionNav.removeClass(disabledClass).filter('.' + namespace +
"prev").addClass(disabledClass).attr('tabindex', '-1');           } else
if (slider.animatingTo === slider.last) {
slider.directionNav.removeClass(disabledClass).filter('.' + namespace +
"next").addClass(disabledClass).attr('tabindex', '-1');

} else {

slider.directionNav.removeClass(disabledClass).removeAttr('tabindex');

}

} else {

slider.directionNav.removeClass(disabledClass).removeAttr('tabindex');

}
}
}

```

```

        },
        pausePlay:
{
    setup: function ()
{
    var pausePlayScaffold = $('<div class="' + namespace +
'pauseplay"><a></a></div>');

    // CONTROLSCONTAINER:
if (slider.controlsContainer) {
    slider.controlsContainer.append(pausePlayS
caffold);           slider.pausePlay =
$('. ' + namespace + 'pauseplay a',
slider.controlsContainer);
} else {           slider.append(pausePlayScaffold);
    slider.pausePlay = $('. ' + namespace + 'pauseplay a', slider);
}
}

methods.pausePlay.update((slider.vars.slideshow) ? namespace + 'pause' :
namespace + 'play');

slider.pausePlay.bind(eventType, function (event) {
event.preventDefault();

if (watchedEvent === "" || watchedEvent === event.type) {
if ($(this).hasClass(namespace + 'pause')) {
    slider.manualPause = true;           slider.manualPlay = false;
    slider.pause();
} else {
    slider.manualPause = false;
    slider.manualPlay = true;
    slider.play();
}
}
}

```

```

        // setup flags to prevent event duplication

if (watchedEvent === "") {
    watchedEvent = event.type;
}

methods.setToClearWatchedEvent();

});

},
update: function (state) {
    (state === "play") ? slider.pausePlay.removeClass(namespace +
'pause').addClass(namespace + 'play').html(slider.vars.playText) :
slider.pausePlay.removeClass(namespace + 'play').addClass(namespace +
'pause').html(slider.vars.pauseText);

}
}, touch:

function () {
    var
    startX,
    startY,
    offset,
    cwidth,
    dx,
    startT,
    scrolling = false,
    localX = 0,
    localY = 0,
    accDx = 0;

    if (!msGesture) {
el.addEventListener('touchstart', onTouchStart, false);

        function onTouchStart(e) {
if (slider.animating) {
e.preventDefault();
} else if (( window.navigator.msPointerEnabled ) || e.touches.length
==== 1) {
            slider.pause(); // 
CAROUSEL: cwidth = (vertical) ?

```

```

slider.h : slider.w;           startT =
Number(new Date());           // CAROUSEL:

// Local vars for X and Y points.

localX = e.touches[0].pageX;      localY =
e.touches[0].pageY;

offset = (carousel && reverse && slider.animatingTo ===
slider.last) ? 0 :
(carousel && reverse) ? slider.limit - (((slider.itemW +
slider.vars.itemMargin) * slider.move) * slider.animatingTo) :
(carousel && slider.currentSlide === slider.last) ? slider.limit :
(carousel) ? ((slider.itemW + slider.vars.itemMargin) * slider.move) *
slider.currentSlide :
(reverse) ? (slider.last - slider.currentSlide +
slider.cloneOffset) * cwidth : (slider.currentSlide + slider.cloneOffset) *
cwidth;
startX = (vertical) ? localY : localX;
startY = (vertical) ? localX : localY;

el.addEventListener('touchmove', onTouchMove, false);
el.addEventListener('touchend', onTouchEnd, false);
}

}

function onTouchMove(e) {
// Local vars for X and Y points.

localX = e.touches[0].pageX;
localY = e.touches[0].pageY;

dx = (vertical) ? startX - localY : startX - localX;
scrolling = (vertical) ? (Math.abs(dx) < Math.abs(localX - startY)) : (Math.abs(dx) < Math.abs(localY - startY));

```

```

var fxms = 500;

if (!scrolling || Number(new Date()) - startT > fxms) {
  e.preventDefault(); if (!fade && slider.transitions)
  {
    if (!slider.vars.animationLoop) { dx =
      dx / ((slider.currentSlide === 0 && dx < 0 || slider.currentSlide === slider.last
      && dx > 0) ? (Math.abs(dx) / cwidth + 2) : 1);
    }
    slider.setProps(offset + dx, "setTouch");
  }
}

function onTouchEnd(e) {
  // finish the touch by undoing the touch session
  el.removeEventListener('touchmove', onTouchMove, false);

  if (slider.animatingTo === slider.currentSlide && !scrolling && !(dx
  === null)) { var updateDx = (reverse) ? -dx : dx,
  target = (updateDx > 0) ? slider.getTarget('next') :
  slider.getTarget('prev');

    if (slider.canAdvance(target) && (Number(new Date()) - startT <
    550 && Math.abs(updateDx) > 50 || Math.abs(updateDx) > cwidth / 2)) {
      slider.flexAnimate(target, slider.vars.pauseOnAction);
    } else { if (!fade)
      slider.flexAnimate(slider.currentSlide, slider.vars.pauseOnAction,
      true);
    }
  }
  el.removeEventListener('touchend', onTouchEnd, false);
}

```

```

        startX = null;
startY = null;
dx = null;           offset
= null;
}

} else {           el.style.msTouchAction = "none";
el._gesture = new MSGesture();           el._gesture.target = el;
el.addEventListener("MSPointerDown", onMSPointerDown, false);
el._slider = slider;
el.addEventListener("MSGestureChange", onMSGestureChange, false);
el.addEventListener("MSGestureEnd", onMSGestureEnd, false);

function onMSPointerDown(e) {
e.stopPropagation();           if (slider.animating)
{
            e.preventDefault();           }
else {           slider.pause();
el._gesture.addPointer(e.pointerId);
accDx = 0;           cwidth = (vertical) ?
slider.h : slider.w;           startT =
Number(new Date());           // CAROUSEL:

offset = (carousel && reverse && slider.animatingTo ===
slider.last) ? 0 :
(carousel && reverse) ? slider.limit - (((slider.itemW +
slider.vars.itemMargin) * slider.move) * slider.animatingTo) :
(carousel && slider.currentSlide === slider.last) ? slider.limit :
(carousel) ? ((slider.itemW + slider.vars.itemMargin) * slider.move) *
slider.currentSlide :
(reverse) ? (slider.last - slider.currentSlide +
slider.cloneOffset) * cwidth : (slider.currentSlide + slider.cloneOffset) * cwidth;
}
}
}

```

```

        function onMSGestureChange(e) {
            e.stopPropagation();
            var slider = e.target._slider;
            if (!slider) { return; }
            var transX = -e.translationX,
            transY = -e.translationY;

            //Accumulate translations. accDx = accDx +
            ((vertical) ? transY : transX); dx = accDx;
            scrolling = (vertical) ? (Math.abs(accDx) < Math.abs(-transX)) :
            (Math.abs(accDx) < Math.abs(-transY));

            if (e.detail === e.MSGESTURE_FLAG_INERTIA) {
                setImmediate(function () {
                    el._gesture.stop();
                });
            }

            return;
        }

        if (!scrolling || Number(new Date()) - startT > 500) {
            e.preventDefault();
            if (!fade && slider.transitions) {
                if (!slider.vars.animationLoop) {
                    dx = accDx /
                    ((slider.currentSlide === 0 && accDx < 0 || slider.currentSlide === slider.last &&
                    accDx > 0) ? (Math.abs(accDx) / cwidth + 2) :
                    1);
                }
                slider.setProps(offset + dx, "setTouch");
            }
        }
    }
}

```

```

        function onMSGestureEnd(e) {
e.stopPropagation();           var
slider = e.target._slider;      if
(!slider) {                  return;
}
if (slider.animatingTo === slider.currentSlide && !scrolling && !(dx
== null)) {                  var updateDx = (reverse) ? -dx : dx,
target = (updateDx > 0) ? slider.getTarget('next') :
slider.getTarget('prev');

if (slider.canAdvance(target) && (Number(new Date()) - startT <
550 && Math.abs(updateDx) > 50 || Math.abs(updateDx) > cwidth / 2)) {
    slider.flexAnimate(target, slider.vars.pauseOnAction);
} else {                      if (!fade)
    slider.flexAnimate(slider.currentSlide, slider.vars.pauseOnAction,
true);
}
startX =
null;                      startY =
null;                      dx =
null;                      offset =
null;                      accDx
= 0;
}
},
},
resize: function () {          if
(!slider.animating && slider.is(':visible')) {
if (!carousel) slider.doMath();

if (fade) {
    // SMOOTH HEIGHT:
methods.smoothHeight();      } else if

```

```

(carousel) { //CAROUSEL:
    slider.slides.width(slider.computedW);
    slider.update(slider.pagingCount);
    slider.setProps();
}

else if (vertical) { //VERTICAL:
    slider.viewport.height(slider.h);
    slider.setProps(slider.h, "setTotal");
} else {
    // SMOOTH HEIGHT:                                     if
    (slider.vars.smoothHeight)                         methods.smoothHeight();
    slider.newSlides.width(slider.computedW);
    slider.setProps(slider.computedW, "setTotal");
}
},
smoothHeight: function (dur) {           if
(!vertical || fade) {                 var $obj = (fade) ? slider :
    slider.viewport;                  (dur) ?
    $obj.animate({ "height": slider.slides.eq(slider.animatingTo).height()}, dur) :
    $obj.height(slider.slides.eq(slider.animatingTo).height())
;
}
},
sync: function (action) {           var $obj =
$(slider.vars.sync).data("flexslider"),           target =
slider.animatingTo;

switch (action) {
case "animate":
    $obj.flexAnimate(target, slider.vars.pauseOnAction, false, true);
break;
case "play":
    if (!$obj.playing && !$obj.asNav) {

```

```

        $obj.play();
    }

break;           case
"pause":         $obj.pause();
break;
}

},
uniqueID: function ($clone) {
$clone.find('[id]').each(function () {
var $this = $(this);
$this.attr('id', $this.attr('id') + '_clone');
});
return $clone;
},
pauseInvisible: {
visProp: null,      init: function () {
var prefixes = ['webkit', 'moz', 'ms', 'o'];

if ('hidden' in document) return 'hidden';          for (var
i = 0; i < prefixes.length; i++) {                  if ((prefixes[i] + 'Hidden')
in document)                      methods.pauseInvisible.visProp =
prefixes[i] + 'Hidden';
}
if (methods.pauseInvisible.visProp) {
var evtname = methods.pauseInvisible.visProp.replace(/\[H|h]idden/, "") +
'visibilitychange';
document.addEventListener(evtname, function () {
if (methods.pauseInvisible.isHidden()) {
if (slider.startTimeout) clearTimeout(slider.startTimeout); //If
clock is ticking, stop timer and prevent from starting while invisible
else slider.pause(); //Or just pause
}
else { if (slider.started)
slider.play(); //Initiated before, just play
else
}
}
}
}
}

```

```

        (slider.vars.initDelay > 0) ? setTimeout(slider.play, slider.vars.initDelay) :
        slider.play(); //Didn't init before: simply init or wait for it
    }
});

}

},
isHidden: function () {
    return
document[methods.pauseInvisible.visProp] || false;
}
},
setToClearWatchedEvent: function () {
clearTimeout(watchedEventClearTimer);
watchedEventClearTimer
= setTimeout(function () {
watchedEvent = "";
}, 3000);
}
};

// public methods
slider.flexAnimate = function (target, pause, override, withSync, fromNav) {
if (!slider.vars.animationLoop && target !== slider.currentSlide) {
slider.direction = (target > slider.currentSlide) ? "next" : "prev";
}

if (asNav && slider.pagingCount === 1) slider.direction = (slider.currentItem
< target) ? "next" : "prev";

if (!slider.animating && (slider.canAdvance(target, fromNav) || override) &&
slider.is(":visible")) {
if (asNav && withSync) {
var master =
$(slider.vars.asNavFor).data('flexslider');
slider.atEnd = target === 0 ||
target === slider.count - 1;
master.flexAnimate(target, true, false, true,
fromNav);
slider.direction = (slider.currentItem < target) ? "next" :
"prev";
master.direction = slider.direction;
}
}
}

```

```

        if (Math.ceil((target + 1) / slider.visible) - 1 !== slider.currentSlide &&
target !== 0) {           slider.currentItem = target;
slider.slides.removeClass(namespace +
"activeslide").eq(target).addClass(namespace + "active-slide");
target = Math.floor(target / slider.visible);
} else {           slider.currentItem = target;
slider.slides.removeClass(namespace +
"activeslide").eq(target).addClass(namespace + "active-
slide");
return false;
}
}

slider.animating = true;
slider.animatingTo = target;

// SLIDESHOW:
if (pause) slider.pause();

// API: before() animation Callback
slider.vars.before(slider);

// SYNC:
if (slider.syncExists && !fromNav) methods.sync("animate");

// CONTROLNAV           if (slider.vars.controlNav)
methods.controlNav.active();

// !CAROUSEL:
// CANDIDATE: slide active class (for add/remove slide)
if (!carousel) slider.slides.removeClass(namespace +
'activeslide').eq(target).addClass(namespace + 'active-slide');

```

```

// INFINITE LOOP:          // CANDIDATE:
atEnd      slider.atEnd = target === 0 || target ===
slider.last;

// DIRECTIONNAV:           if (slider.vars.directionNav)
methods.directionNav.update();

if (target === slider.last) {
// API: end() of cycle Callback
slider.vars.end(slider);

// SLIDESHOW && !INFINITE LOOP:
if (!slider.vars.animationLoop) slider.pause();
}

// SLIDE:                  if (!fade) {           var dimension =
(vertical) ? slider.slides.filter(':first').height() : slider.computedW,
margin, slideString, calcNext;

// INFINITE LOOP / REVERSE:
if (carousel) {
//margin = (slider.vars.itemWidth > slider.w) ? slider.vars.itemMargin
* 2 : slider.vars.itemMargin;           margin =
slider.vars.itemMargin;           calcNext = ((slider.itemW
+ margin) * slider.move) * slider.animatingTo;
slideString = (calcNext > slider.limit && slider.visible !== 1) ?
slider.limit : calcNext;
} else if (slider.currentSlide === 0 && target === slider.count - 1 &&
slider.vars.animationLoop && slider.direction !== "next") {
slideString = (reverse) ? (slider.count + slider.cloneOffset) * dimension : 0;
} else if (slider.currentSlide === slider.last && target === 0 &&
slider.vars.animationLoop && slider.direction !== "prev") {
slideString = (reverse) ? 0 : (slider.count + 1) * dimension;
} else {           slideString = (reverse) ? ((slider.count - 1) - target
+ slider.cloneOffset)
}
}

```

```

* dimension : (target + slider.cloneOffset) * dimension;
}

slider.setProps(slideString, "", slider.vars.animationSpeed);

if (slider.transitions) { if (!slider.vars.animationLoop ||
!slider.atEnd) { slider.animating = false;
slider.currentSlide = slider.animatingTo;
}

slider.container.unbind("webkitTransitionEnd transitionend");

slider.container.bind("webkitTransitionEnd transitionend", function ()
{
    slider.wrapup(dimension);
});

} else { slider.container.animate(slider.args,
slider.vars.animationSpeed, slider.vars.easing, function () {
    slider.wrapup(dimension);
});

}

} else { // FADE:
if (!touch) {
//slider.slides.eq(slider.curre
ntSlide).fadeOut(slider.vars.a
imationSpeed,
slider.vars.easing);
//slider.slides.eq(target).fadeIn(slider.vars.animationSpeed,
slider.vars.easing, slider.wrapup);

slider.slides.eq(slider.currentSlide).css({ "zIndex": 1 }).animate({ "opacity": 0 }, slider.vars.animationSpeed, slider.vars.easing);
slider.slides.eq(target).css({ "zIndex": 2 }).animate({ "opacity": 1 },
slider.vars.animationSpeed, slider.vars.easing, slider.wrapup);

} else {
slider.slides.eq(slider.currentSlide).css({ "opacity": 0, "zIndex": 1 });

```

```

slider.slides.eq(target).css({ "opacity": 1, "zIndex": 2});
slider.wrapup(dimension);

}

}

// SMOOTH HEIGHT:

if (slider.vars.smoothHeight)

methods.smoothHeight(slider.vars.animationSpeed);

}

};

slider.wrapup = function (dimension) {

// SLIDE:

if (!fade && !carousel) {

if (slider.currentSlide === 0 && slider.animatingTo === slider.last &&
slider.vars.animationLoop) { slider.setProps(dimension, "jumpEnd");

} else if (slider.currentSlide === slider.last && slider.animatingTo === 0
&& slider.vars.animationLoop) {

slider.setProps(dimension, "jumpStart");

}

}

slider.animating = false;
slider.currentSlide = slider.animatingTo;

// API: after() animation Callback

slider.vars.after(slider);

};




// SLIDESHOW:

slider.animateSlides = function () {

if (!slider.animating && focused)
slider.flexAnimate(slider.getTarget("next"))

;

};




// SLIDESHOW:

slider.pause = function () {

clearInterval(slider.animatedSlides); slider.animatedSlides =

```

```

null;      slider.playing = false;      // PAUSEPLAY:
if (slider.vars.pausePlay) methods.pausePlay.update("play");
// SYNC:
if (slider.syncExists) methods.sync("pause");
};

// SLIDESHOW:
slider.play = function () { if (slider.playing)
clearInterval(slider.animatedSlides);      slider.animatedSlides =
slider.animatedSlides || setInterval(slider.animateSlides,
slider.vars.slideshowSpeed);      slider.started = slider.playing =
true;      // PAUSEPLAY:      if (slider.vars.pausePlay)
methods.pausePlay.update("pause");      // SYNC:
if (slider.syncExists) methods.sync("play");
};

// STOP:
slider.stop = function () { slider.pause();
slider.stopped = true;
};

slider.canAdvance = function (target, fromNav) {
// ASNAV:      var last = (asNav) ? slider.pagingCount - 1
: slider.last;      return (fromNav) ? true :
(asNav && slider.currentItem === slider.count - 1 && target === 0 &&
slider.direction === "prev") ? true :
(asNav && slider.currentItem === 0 && target === slider.pagingCount -
1 && slider.direction !== "next") ? false :
(target === slider.currentSlide && !asNav) ? false :
(slider.vars.animationLoop) ? true :
(slider.atEnd && slider.currentSlide === 0 && target === last
&& slider.direction !== "next") ? false :
(slider.atEnd && slider.currentSlide === last && target === 0
&& slider.direction === "next") ? false :
true;
};

```

```

    slider.getTarget = function (dir) {           slider.direction = dir;           if
    (dir === "next") {           return (slider.currentSlide === slider.last) ? 0 :
    slider.currentSlide + 1;
    } else {           return (slider.currentSlide === 0) ? slider.last :
    slider.currentSlide - 1;
    }
};

// SLIDE:     slider.setProps = function
(pos, special, dur) {           var target = (function
() {
    var posCheck = (pos) ? pos : ((slider.itemW + slider.vars.itemMargin) *
    slider.move) * slider.animatingTo,
    posCalc = (function () {
if (carousel) {
    return (special === "setTouch") ? pos :
    (reverse && slider.animatingTo === slider.last) ? 0 :
    (reverse) ? slider.limit - (((slider.itemW +
    slider.vars.itemMargin) * slider.move) * slider.animatingTo) :
    (slider.animatingTo === slider.last) ? slider.limit : posCheck;
} else {
switch (special) {
case "setTotal":
    return (reverse) ? ((slider.count - 1) - slider.currentSlide +
    slider.cloneOffset) * pos : (slider.currentSlide + slider.cloneOffset) * pos;
case "setTouch":
    return (reverse) ? pos : pos;
case "jumpEnd":
    return (reverse) ? pos : slider.count * pos;
case "jumpStart":
    return (reverse) ? slider.count * pos : pos;
default:
    return pos;
}
}
});
```

```

        }
    }
}());

return (posCalc * -1) + "px";
}());

if (slider.transitions) {           target = (vertical) ? "translate3d(0," + target
+ ",0)" : "translate3d(" + target
+ ",0,0)";           dur = (dur !== undefined) ? (dur / 1000) + "s" : "0s";
slider.container.css("-" + slider.pfx + "-transition-duration", dur);
slider.container.css("transition-duration", dur);
}

slider.args[slider.prop] = target;
if (slider.transitions || dur === undefined) slider.container.css(slider.args);

slider.container.css('transform', target);
};

slider.setup = function (type) {
// SLIDE:      if (!fade) {
var sliderOffset, arr;

if (type === "init") {           slider.viewport =
$('.<div class=' + namespace +
'verport"></div>').css({
"overflow": "hidden",
"position": "relative"
}).appendTo(slider).append(slider.container);
// INFINITE LOOP:           slider.cloneCount = 0;
slider.cloneOffset = 0;           // REVERSE:
if (reverse) {           arr =
$.makeArray(slider.slides).reverse();

```

```

slider.slides = $(arr);
slider.container.empty().append(slider.slides);
}

}

// INFINITE LOOP && !CAROUSEL:
if (slider.vars.animationLoop && !carousel) {

slider.cloneCount = 2;           slider.cloneOffset = 1;
// clear out old clones         if (type !== "init")
slider.container.find('.clone').remove();           //

slider.container.append(slider.slides.first().clone().addClass('clone').attr('aria-hidden',
'true')).prepend(slider.slides.last().clone().addClass('clone').attr('aria-hidden', 'true'));
methods.uniqueID(slider.slides.first().clone().addClass('clone').attr('ariahidden',
'true')).appendTo(slider.container);
methods.uniqueID(slider.slides.last().clone().addClass('clone').attr('ariahidden',
'true')).prependTo(slider.container);
}

slider.newSlides = $(slider.vars.selector, slider);

sliderOffset = (reverse) ? slider.count - 1 - slider.currentSlide +
slider.cloneOffset : slider.currentSlide + slider.cloneOffset;           //

VERTICAL:
if (vertical && !carousel) {

slider.container.height((slider.count + slider.cloneCount) * 200 +
"%").css("position", "absolute").width("100%");
setTimeout(function () {
slider.newSlides.css({"display": "block"});
slider.doMath();           slider.viewport.height(slider.h);
slider.setProps(sliderOffset * slider.h, "init");
}, (type === "init") ? 100 : 0);
} else {           slider.container.width((slider.count +
slider.cloneCount) * 200 + "%");           slider.setProps(sliderOffset *
slider.computedW, "init");
setTimeout(function () {

```

```

slider.doMath();           slider.newSlides.css({"width": slider.computedW,
"float": "left",
"display": "block"}); 

    // SMOOTH HEIGHT:
    if (slider.vars.smoothHeight) methods.smoothHeight();
}, (type === "init") ? 100 : 0);
}

} else { // FADE:
    slider.slides.css({"width": "100%", "float": "left", "marginRight": "-100%",
"position": "relative"});
if (type === "init") {
if (!touch) {

//slider.slides.eq(slider.currentSlide).fadeIn(slider.vars.animationSpeed,
slider.vars.easing);
    slider.slides.css({
"opacity": 0,
"display": "block",
"zIndex": 1
}).eq(slider.currentSlide).css({"zIndex": 2}).animate({"opacity": 1},
slider.vars.animationSpeed, slider.vars.easing);
} else {
    slider.slides.css({
"opacity": 0,
"display": "block",
"webkitTransition": "opacity " + slider.vars.animationSpeed / 1000
+ "s ease",
"zIndex": 1
}).eq(slider.currentSlide).css({"opacity": 1, "zIndex": 2});
}
}

// SMOOTH HEIGHT:
if (slider.vars.smoothHeight) methods.smoothHeight();

```

```

        }

    // !CAROUSEL:

    // CANDIDATE: active slide      if (!carousel)
slider.slides.removeClass(namespace +
"activeslide").eq(slider.currentSlide).addClass(namespace + "active-
slide");

    //FlexSlider: init() Callback
slider.vars.init(slider);

};

    slider.doMath = function () {
        var
slide = slider.slides.first(),
slideMargin = slider.vars.itemMargin,
minItems = slider.vars.minItems,
maxItems = slider.vars.maxItems;

        slider.w = (slider.viewport === undefined) ? slider.width() :
slider.viewport.width();      slider.h = slide.height();
slider.boxPadding = slide.outerWidth() - slide.width();

        // CAROUSEL:      if (carousel) {  slider.itemT =
slider.vars.itemWidth + slideMargin;      slider.minW = (minItems) ?
minItems * slider.itemT : slider.w;      slider.maxW = (maxItems) ?
(maxItems * slider.itemT) - slideMargin : slider.w;      slider.itemW =
(slider.minW > slider.w) ? (slider.w - (slideMargin * (minItems - 1))) / minItems
:
(slider.maxW < slider.w) ? (slider.w - (slideMargin * (maxItems - 1))) /
maxItems :
(slider.vars.itemWidth > slider.w) ? slider.w : slider.vars.itemWidth;

        slider.visible = Math.floor(slider.w / (slider.itemW));      slider.move
= (slider.vars.move > 0 && slider.vars.move < slider.visible ) ? slider.vars.move :
slider.visible;      slider.pagingCount = Math.ceil(((slider.count - slider.visible) /
slider.move)

```

```

+ 1);

        slider.last = slider.pagingCount - 1;
        slider.limit = (slider.pagingCount === 1) ? 0 :
            (slider.vars.itemWidth > slider.w) ? (slider.itemW * (slider.count - 1)) +
            (slideMargin * (slider.count - 1)) : ((slider.itemW + slideMargin) * slider.count) -
            slider.w - slideMargin;

    } else {
        slider.itemW =
        slider.w;           slider.pagingCount =
        slider.count;       slider.last =
        slider.count - 1;
    }

    slider.computedW = slider.itemW - slider.boxPadding;
};

slider.update = function (pos, action) {
    slider.doMath();

    // update currentSlide and slider.animatingTo if necessary
    if (!carousel) {
        if (pos < slider.currentSlide) {
            slider.currentSlide += 1;
        } else if (pos <= slider.currentSlide && pos !== 0) {
            slider.currentSlide -= 1;
        }
        slider.animatingTo = slider.currentSlide;
    }

    // update controlNav
    if (slider.vars.controlNav &&
    !slider.manualControls) {
        if ((action === "add" &&
    !carousel) || slider.pagingCount > slider.controlNav.length) {
            methods.controlNav.update("add");
        }
    }
}

```

Apply.js:

```
import {  
  SafeAreaView,  
  TextInput,  
  Text,  
  TouchableOpacity,  
  View,  
  Button,  
  Linking  
} from "react-native";  
  
import React from "react";  
  
import Spacing from "../constants1/Spacing";  
  
import FontSize from "../constants1/FontSize";  
  
import Colors from "../constants1/Colors";  
  
import Font from "../constants1/Font";  
  
import { Ionicons } from "@expo/vector-icons";  
  
import { AntDesign } from '@expo/vector-icons';  
  
import { useState } from "react";  
  
import { useEffect } from "react";  
  
import * as MailComposer from "expo-mail-composer";  
400
```

```
import * as Print from "expo-print";

import * as Permissions from 'expo-permissions';

import { useNavigation } from "@react-navigation/native";



const Applyforum = () => {

  const navigation = useNavigation();

  const [name, setName] = useState("");

  const [focused, setFocused] = useState(false);

  const [isAvailable, setIsAvailable] = useState(false);

  const [recipients, setRecipients] = useState([]);

  const [subject, setSubject] = useState(undefined);

  const [body, setBody] = useState(undefined);

  const [email, setEmail] = useState(undefined);




  useEffect(() => {

    async function checkAvailability() {

      const isMailAvailable = await MailComposer.isAvailableAsync();

      setIsAvailable(isMailAvailable);

    }

  })

}
```

```
    checkAvailability();

}, []);

// const sendMail = async () => {

//   const { uri } = await Print.printToFileAsync({

//     html: "<h1>My pdf!</h1>"

//   });

//   // const { status } = await Permissions.askAsync(Permissions.READ_CONTACTS);

//   // if (status !== 'granted') {

//     // console.error('Contacts permission not granted');

//     // return;

//   // }

//   try {

//     MailComposer.composeAsync({

//       subject: subject,

//       body: body,

//       recipients: recipients,

//       attachments: [uri]

//     });

//   } catch(error) {
```

```
//   console.log('Error sending email:', error); // Log the detailed error

// }

// };

const sendMail = () => {

  Linking.openURL(`mailto:${recipients.join(",")}?subject=${subject}&body=${body}`);

}

const addRecipient = () => {

  let newRecipients = [...recipients];

  newRecipients.push(email);

  setRecipients(newRecipients);

  setEmail(undefined);

};

const showRecipients = () => {

  if (recipients.length === 0) {

    return <Text>No recipients added</Text>;

  }

  return recipients.map((recipient, index) => {
```

```
        return <Text key={index}>{recipient}</Text>;\n\n    });\n\n};\n\nreturn (\n\n    <SafeAreaView>\n\n        <View\n            style={{\n                padding: Spacing * 2,\n                backgroundColor: 'white',\n                height: '100%',\n            }}\n\n        >\n\n        <View\n            style={{\n                alignItems: "center",\n                marginTop: '20px',\n            }}\n\n        >\n\n        <TouchableOpacity //133\n            style={{\n
```

```
padding: Spacing,  
backgroundColor: Colors.gray,  
borderRadius: Spacing / 2,  
marginHorizontal: Spacing,  
}  
}  
  
onPress={() => navigation.navigate("Search Jobs")}  
  
>  
  
<AntDesign name="arrowleft" size={24} color="black" />  
  
</TouchableOpacity>  
  
</View>  
  
<View  
style={{  
marginVertical: Spacing * 3,  
backgroundColor: Colors.lightPrimary,  
}  
}  
  
>
```

```
<TextInput  
    onFocus={() => setFocused(true)}  
    onBlur={() => setFocused(false)}  
    placeholderTextColor={Colors.darkText}  
    placeholder="Subject"  
    value={subject}  
    onChangeText={setSubject}  
  
    style={[  
        {  
            fontSize: FontSize.small,  
            padding: Spacing * 2,  
            backgroundColor: Colors.lightPrimary,  
            borderRadius: Spacing,  
            marginVertical: Spacing,  
            borderColor: 'black',  
            borderWidth: 1,  
        },  
        focused && {
```

```
borderWidth: 3,  
borderColor: Colors.primary,  
shadowOffset: { width: 4, height: Spacing },  
shadowColor: Colors.primary,  
shadowOpacity: 0.2,  
shadowRadius: Spacing,  
},  
]  
/  
<TextInput  
onFocus={() => setFocused(true)}  
onBlur={() => setFocused(false)}  
placeholderTextColor={Colors.darkText}  
placeholder="Message"  
value={body}  
onChangeText={setBody}  
style={[  
{
```

```
        fontSize: FontSize.small,  
        padding: Spacing * 2,  
        backgroundColor: Colors.lightPrimary,  
        borderRadius: Spacing,  
        marginVertical: Spacing,  
        borderColor: 'black',  
        borderWidth: 1,  
    },  
  
    focused && {  
        borderWidth: 3,  
        borderColor: Colors.primary,  
        shadowOffset: { width: 4, height: Spacing },  
        shadowColor: Colors.primary,  
        shadowOpacity: 0.2,  
        shadowRadius: Spacing,  
    },  
},  
]  
/>
```

```
<TextInput  
    onFocus={() => setFocused(true)}  
    onBlur={() => setFocused(false)}  
    placeholderTextColor={Colors.darkText}  
    placeholder="Email"  
    value={email}  
    onChangeText={setEmail}  
    style={[  
        {  
            fontSize: FontSize.small,  
            padding: Spacing * 2,  
            backgroundColor: Colors.lightPrimary,  
            borderRadius: Spacing,  
            marginVertical: Spacing,  
            borderColor: 'black',  
            borderWidth: 1,  
        },  
    ]}
```

```
focused && {  
  borderWidth: 3,  
  
  borderColor: Colors.primary,  
  
  shadowOffset: { width: 4, height: Spacing },  
  
  shadowColor: Colors.primary,  
  
  shadowOpacity: 0.2,  
  
  shadowRadius: Spacing,  
},  
]  
}
```

/>

```
<Button title='Add Recipient' onPress={addRecipient}/>
```

```
{showRecipients()}
```

</View>

```
<TouchableOpacity
```

```
style={{
```

```
padding: Spacing * 2,  
  
backgroundColor: Colors.primary,  
  
marginVertical: Spacing * 3,  
  
borderRadius: Spacing,  
  
shadowColor: Colors.primary,  
  
shadowOffset: {  
  
  width: 0,  
  
  height: Spacing,  
  
},  
  
shadowOpacity: 0.3,  
  
shadowRadius: Spacing,  
  
}}  
  
onPress={() => sendMail()}  
  
>
```

```
<Text  
  style={{  
  
    color: Colors.onPrimary,  
  
    textAlign: "center",  
  
    fontSize: FontSize.large,  
  
  }}  
  
>
```

Send Email

</Text>

</TouchableOpacity>

<View

style={{

marginVertical: Spacing * 3,

}}

>

<View

style={{

marginTop: Spacing,

flexDirection: "row",

justifyContent: "center",

}}

>

```
</View>

</View>

</View>

</SafeAreaView>

);

};

export default Applyforum;
```

Description Job.js:

```
import * as React from "react";

import { Image, StyleSheet, Pressable, View, Text, Alert, FlatList, ScrollView, Button } from
"react-native";

//import LinearGradient from "react-native-linear-gradient";

import { LinearGradient } from "expo-linear-gradient";
413
```

```
import { useNavigation } from "@react-navigation/native";

import { Color, Padding, FontSize, Border } from "../GlobalStyles";

import { useRoute } from "@react-navigation/native";

import * as Location from 'expo-location';

import * as IntentLauncher from 'expo-intent-launcher';

import { Linking } from 'react-native';

const Job_des = () => {

    const navigation = useNavigation();

    const route = useRoute();

    try {

        const {

            job_name,

            company,

            salary,

            image,

            applybefore,

            jobtype,

            posted_on,

            job_description,
```

```
job_qualityscore,  
job_applylink,  
longitude,  
latitude,  
website,  
} = route.params;  
  
const handleUrl = async () => {  
  
    const url = await website;  
  
    Linking.openURL(url);  
  
}  
  
const handlePress = async () => {  
  
    // Check if location services are enabled  
  
    let { status } = await Location.requestForegroundPermissionsAsync();  
  
    if (status !== 'granted') {  
  
        Alert.alert('Location permission not granted', 'Please enable location services to find  
nearby hospitals.');  
    }  
  
    return;
```

```
}

// // Get current location

// let location = await Location.getCurrentPositionAsync({ });

// const { latitude, longitude } = location.coords;

// Form Google Maps URL with current location

console.log(latitude);

console.log(longitude);

const mapsUrl = `https://www.google.com/maps?q=${latitude},${longitude}`;

// Open Google Maps or fallback to browser

try {

    await IntentLauncher.startActivityAsync('android.intent.action.VIEW', {

        data: mapsUrl,

    });

} catch (error) {

    console.error('Error opening Google Maps:', error);

    Alert.alert('Error', 'Failed to open Google Maps. Please try again later.');

}

};
```

```
return (
```



```
<View style={styles.screen}>
```



```
    <View style={[styles.frameParent, styles.parentPosition1]}>
```



```
        <ScrollView style ={{
```



```
            backgroundColor: Color.primaryWHITE,
```



```
            maxHeight: 5000,
```



```
        }}>
```



```
        <View style={styles.frameGroup}>
```



```
            <View style={[styles.arrowBack24pxParent, styles.parentFlexBox]}>
```



```
                <Pressable
```



```
                    style={[styles.arrowBack24px, styles.arrowBack24pxLayout]}
```



```
                    onPress={() => navigation.navigate("Jobs3")}
```



```
                >
```

```
<Image
    style={styles.icon}
    resizeMode="cover"
    source={require("../assets/arrow_back_24px.png")}

/>

</Pressable>

{/* <Image
    style={[
        styles.favoriteBorder24pxIcon,
        styles.arrowBack24pxLayout,
    ]}

    resizeMode="cover"
    source={require("../assets/favorite_border_24px.png")}

/> */}

<Pressable onPress={() => navigation.navigate("Applyforum")}>

<View style={styles.applyNowWrapper}>
    <Text style={[styles.applyNow, styles.applyTypo]}>
        APPLY NOW
    </Text>
</View>

```

```
</Text>

</View>

</Pressable>

</View>

<View style={styles.image3Parent}>

<Pressable onPress= {handleUrl}>

<Image

style={styles.image3Icon}

resizeMode="cover"

source={{ uri: image }}

/>

</Pressable>

<View style={styles.midLevelUXDesignerParent}>

<Text style={[styles.midLevelUXDesigner, styles.toptalFlexBox]}>

{job_name}

</Text>

<Text style={[styles.toptal, styles.toptalTypo]}>

{company}
```

```
</Text>

<Text style={[styles.postedOn20, styles.july2021Typo]}>

  Posted on {posted_on}

</Text>

<Text>

  Job Quality Score: {job_qualityscore}

</Text>

<Text>

  Job Apply Link: {job_applylink}

</Text>

</View>

</View>

<View style={[styles.frameWrapper, styles.frameWrapperSpaceBlock]}>

  <View style={styles.frameContainer}>

    <View style={[styles.applyBeforeParent, styles.parentPosition1]}>

      <Text style={[styles.applyBefore, styles.applyTypo]}>

        Apply Before

      </Text>
    
```

```
<Text style={[styles.july2021, styles.july2021Typo]}>  
    {applybefore}  
</Text>  
  
</View>  
  
<View style={[styles.jobNatureParent, styles.parentPosition1]}>  
  
<Text style={[styles.applyBefore, styles.applyTypo]}>  
    Job Nature  
</Text>  
  
<View style={styles.frameView}>  
  
<View style={styles.fullTimeWrapper}>  
  
<Text style={[styles.fullTime, styles.fullTimeLayout]}>  
    Concenture  
</Text>  
  
</View>  
  
<View style={styles.frameChild} />  
  
</View>  
  
</View>  
  
<View style={[styles.salaryRangeParent, styles.parentPosition]}>  
  
<Text style={[styles.applyBefore, styles.applyTypo]}>  
    Job City  
</Text>
```

```
<Text style={[styles.july2021, styles.july2021Typo]}>  
    {salary}  
</Text>  
  
</View>  
  
<View style={[styles.jobLocationParent, styles.parentPosition]}>  
  
<Text style={[styles.applyBefore, styles.applyTypo]}>  
    Job Type  
</Text>  
  
<Text style={[styles.july2021, styles.july2021Typo]}>  
    {jobtype}  
</Text>  
  
</View>  
  
</View>  
  
</View>  
  
<View style={[styles.frameWrapper1, styles.frameWrapperSpaceBlock]}>  
  
<View style={styles.frameParent1}>  
  
<LinearGradient
```

```
    style={styles.jobDescriptionParent}

    locations={[0, 1]}

    colors={"#fff", "rgba(255, 255, 255, 0)"}

    useAngle={true}

    angle={180}

>

<Text style={[styles.applyBefore, styles.applyTypo]}>

  Job Description

</Text>

<Text style={[styles.canYouBring, styles.july2021Typo]}>

  {job_description}

</Text>

</LinearGradient>

</View>

</View>

</ScrollView>

/* <View style={[styles.frameWrapper2, styles.frameWrapperSpaceBlock]}>
```

```
<View>
```

```
<View style={styles.applyNowWrapper}>

    <Text style={[styles.applyNow, styles.applyTypo]}>
        APPLY NOW
    </Text>

</View>

</View> */}

</View>

</View>

);

} catch (e) {
    console.log(e);
    return Alert.alert(
        "No item selected to display",
        "Choose options",
        [
            {
                text: "Ask me later",
                onPress: () => console.log("Ask me later pressed"),
            }
        ]
    );
}
```

```
        },
        {
          text: "Cancel",
          onPress: () => console.log("Cancel Pressed"),
          style: "cancel",
        },
      ],
      { text: "OK", onPress: () => navigation.navigate("Jobs3") },
    );
  };
};

const styles = StyleSheet.create({
  parentPosition1: {
    top: 0,
    position: "absolute",
  },
  parentFlexBox: {
    alignItems: "center",
  },
});
```

```
},  
  
arrowBack24pxLayout: {  
  
  height: 24,  
  
  width: 24,  
  
  position: "absolute",  
  
},  
  
toptalFlexBox: {  
  
  textAlign: "left",  
  
  color: Color.textColor,  
  
},  
  
toptalTypo: {  
  
  opacity: 0.8,  
  
  //fontFamily: FontFamily.poppinsMedium,  
  
  fontWeight: "500",  
  
},  
  
july2021Typo: {  
  
  //fontFamily: FontFamily.poppinsRegular,  
  
  textAlign: "left",  
  
},  
  
frameWrapperSpaceBlock: {  
  
  padding: Padding.p_5xl,
```

```
marginTop: 10,  
  
backgroundColor: Color.primaryWHITE,  
  
},  
  
applyTypo: {  
  
//fontFamily: FontFamily.poppinsBold,  
  
fontWeight: "700",  
  
letterSpacing: 1,  
  
textAlign: "left",  
  
},  
  
fullTimeLayout: {  
  
lineHeight: 13,  
  
fontSize: FontSize.aPPLabel12M_size,  
  
color: Color.textColor,  
  
},  
  
parentPosition: {  
  
top: 61,  
  
position: "absolute",  
  
},  
  
seeTypo: {  
  
color: Color.primary,  
  
// fontFamily: FontFamily.poppinsSemiBold,
```

```
fontWeight: "600",  
letterSpacing: 0,  
textAlign: "left",  
,  
icon: {  
height: "100%",  
width: "100%",  
,  
arrowBack24px: {  
left: -6,  
top: 27,  
zIndex: 0,  
,  
favoriteBorder24pxIcon: {  
top: 3,  
left: 339,  
zIndex: 1,  
,  
arrowBack24pxParent: {  
alignSelf: "stretch",  
justifyContent: "space-between",
```

```
        },  
  
    image3Icon: {  
  
        borderRadius: Border.br_9xs,  
  
        width: 64,  
  
        height: 64,  
  
    },  
  
    midLevelUxDesigner: {  
  
        fontSize: FontSize.size_7xl,  
  
        lineHeight: 31,  
  
        //fontFamily: fontFamily.poppinsMedium,  
  
        fontWeight: "500",  
  
        textAlign: "left",  
  
    },  
  
    toptal: {  
  
        marginTop: 6,  
  
        lineHeight: 15,  
  
        fontSize: FontSize.aPP14Medium_size,  
  
        textAlign: "left",  
  
        color: Color.textColor,  
  
    },  
  
    postedOn20: {
```

```
    opacity: 0.6,  
  
    lineHeight: 13,  
  
    fontSize: FontSize.aPPLabel12M_size,  
  
    color: Color.textColor,  
  
    marginTop: 6,  
  
,  
  
midLevelUxDesignerParent: {  
  
    marginTop: 16,  
  
,  
  
image3Parent: {  
  
    marginTop: 24,  
  
,  
  
frameGroup: {  
  
    paddingHorizontal: Padding.p_5xl,  
  
    paddingTop: Padding.p_11xl,  
  
    paddingBottom: Padding.p_xl,  
  
    width: 411,  
  
    backgroundColor: Color.primaryWHITE,  
  
,  
  
applyBefore: {  
  
    fontSize: FontSize.size_xs_5,
```

```
lineHeight: 12,  
textTransform: "uppercase",  
opacity: 0.75,  
color: Color.textColor,  
,  
july2021: {  
marginTop: 10,  
lineHeight: 15,  
fontSize: FontSize.aPP14Medium_size,  
color: Color.textColor,  
,  
applyBeforeParent: {  
left: 0,  
,  
fullTime: {  
textAlign: "right",  
opacity: 0.8,  
//fontFamily: FontFamily.poppinsMedium,  
fontWeight: "500",  
,  
fullTimeWrapper: {
```

```
paddingHorizontal: Padding.p_5xs,  
paddingVertical: Padding.p_9xs,  
borderRadius: Border.br_6xl,  
flexDirection: "row",  
backgroundColor: Color.bG,  
},  
  
frameChild: {  
backgroundColor: Color.colorGhostwhite_100,  
display: "none",  
marginLeft: 8,  
borderRadius: Border.br_6xl,  
},  
  
frameView: {  
marginTop: 8,  
flexDirection: "row",  
},  
  
jobNatureParent: {  
left: 188,  
},  
  
salaryRangeParent: {  
left: 0,
```

```
    },  
  
    jobLocationParent: {  
  
        left: 188,  
  
    },  
  
    frameContainer: {  
  
        width: 334,  
  
        height: 98,  
  
    },  
  
    frameWrapper: {  
  
        marginTop: 10,  
  
        width: 411,  
  
    },  
  
    canYouBring: {  
  
        lineHeight: 20,  
  
        color: Color.primaryBLACK,  
  
        width: 363,  
  
        marginTop: 12,  
  
        fontSize: FontSize.aPP14Medium_size,  
  
    },  
  
    jobDescriptionParent: {  
  
        backgroundColor: "transparent",
```

```
    },  
  
    seeMore: {  
  
        fontSize: FontSize.size_smi,  
  
    },  
  
    arrowDropDown24px: {  
  
        width: 16,  
  
        height: 16,  
  
        marginLeft: 4,  
  
    },  
  
    seeMoreParent: {  
  
        marginTop: 16,  
  
    },  
  
    frameParent1: {  
  
        backgroundColor: Color.primaryWHITE,  
  
    },  
  
    frameWrapper1: {  
  
        marginTop: 10,  
  
    },  
  
    seeMore1: {  
  
        fontSize: FontSize.aPPLabel12M_size,  
  
        color: Color.primary,  

```

```
//fontFamily: FontFamily.poppinsSemiBold,  
  
fontWeight: "600",  
  
letterSpacing: 0,  
  
},  
  
applyNow: {  
  
marginLeft: -45,  
  
top: 18,  
  
left: "50%",  
  
fontSize: FontSize.sFSubheadlineSemibold_size,  
  
color: Color.primaryWHITE,  
  
width: 97,  
  
height: 15,  
  
lineHeight: 15,  
  
position: "absolute",  
  
fontWeight: "700",  
  
letterSpacing: 1,  
  
},  
  
applyNowWrapper: {  
  
borderRadius: Border.br_xs,  
  
backgroundColor: Color.primary,  
  
shadowColor: "rgba(28, 88, 242, 0.1)",
```

```
shadowOffset: {
```

```
  width: 0,
```

```
  height: 4,
```

```
},
```

```
shadowRadius: 24,
```

```
elevation: 24,
```

```
shadowOpacity: 1,
```

```
width: 250,
```

```
height: 52,
```

```
marginTop: 16,
```

```
marginLeft: 90,
```

```
},
```

```
frameWrapper2: {
```

```
  height: 175,
```

```
  marginTop: 10,
```

```
},
```

```
frameParent: {
```

```
  width: 430,
```

```
  left: 0,
```

```
  height: 943,
```

```
},
```

```
screen: {  
  flex: 1,  
  
  overflow: "hidden",  
  
  height: 943,  
  
  width: "100%",  
  
  // backgroundColor: 'black',  
  
},  
});  
  
  
  
  
export default Job_des;
```

Submit Post.js:

```
import React, { useState, useContext, useEffect } from 'react';  
  
import { View, Text, Platform, StyleSheet, Alert,  
  
ActivityIndicator, Button, TextInput, Image } from 'react-native';  
  
import * as ImagePicker from 'expo-image-picker';  
  
// import * as firebase from 'firebase';  
  
import { firebase } from "../config.firebaseio";  
  
import { StatusWrapper, SubmitBtn,  
  
SubmitBtnText, InputWrapper, InputField } from "../styles/AddPost";  
437
```

```
import { LinearGradient } from 'expo-linear-gradient';

import { ScrollView } from 'react-native-gesture-handler';

import { useNavigation } from "@react-navigation/native";

import * as FileSystem from 'expo-file-system';

const Addpostscreen = ({ route }) => {

  const { fetchPosts } = route.params;

  // const { user } = useContext(AuthContext);

  const navigation = useNavigation();

  const [currentUserName, setCurrentUserName] = useState("");

  const [profileimageuri, profilesetImageuri] = useState();

  useEffect(() =>

  {

    firebase.firestore().collection('users')

      .doc(firebase.auth().currentUser.uid)

      .get()

      .then((documentSnapshot) => {

        if (documentSnapshot.exists) {

          setCurrentUserName(documentSnapshot.data().name);


```

```
profilesetImageuri(documentSnapshot.data().imageuri);

}

else{
    console.log('User does not exist');

}

});

const currentUserId = firebase.auth().currentUser.uid;

const [image, setImage] = useState(null);

const [uploading, setUploading] = useState(false);

const [transferred, setTransferred] = useState(0);

const [post, setPost] = useState(null);

const takePhotoFromCamera = async () => {

    let result = await ImagePicker.launchCameraAsync({
        mediaTypes: ImagePicker.MediaTypeOptions.Images,
```

```
        allowsEditing: true,  
  
        aspect: [4, 3],  
  
        quality: 1,  
  
    });  
  
    handleImagePickerResult(result);  
  
};  
  
const choosePhotoFromLibrary = async () => {  
  
    let result = await ImagePicker.launchImageLibraryAsync({  
  
        mediaTypes: ImagePicker.MediaTypeOptions.Images,  
  
        allowsEditing: true,  
  
        aspect: [4, 3],  
  
        quality: 1,  
  
    });  
  
    handleImagePickerResult(result);  
  
};
```

```
const handleImagePickerResult = async (result) => {  
  if (!result.canceled) {  
    if (result.assets && result.assets.length > 0) {  
      setImage(result.assets[0].uri);  
    }  
  }  
};
```

```
const submitPost = async (fetchPostsCallback) => {  
  if (!image) {  
    Alert.alert('Error', 'Please select an image for your post.');//  
    return;  
  }  
  
  if (!post) {  
    Alert.alert('Error', 'Please write text for your post.');//  
    return;  
  }
```

```
const imageUrl = await uploadImage();

console.log('Image Url:', imageUrl);

console.log('Post:', post);

firebase.firestore().collection('posts').add({
    profileimg: profileimageuri,
    userName: currentUserName,
    userId: currentUserId,
    post: post,
    postImg: imageUrl,
    postTime: firebase.firestore.FieldValue.serverTimestamp(),
    likes: 0,
    comments: 0,
})

.then(() => {
    console.log('Post added!');
    Alert.alert(
        'Post published!',
        'Your post has been published Successfully!',
    )
})
```

```
);

setPost(null);

fetchPosts();

})

.catch((error) => {

  console.log('Something went wrong with added post to firestore.', error);

});

setImage(null);

};

const uploadImage = async () => {

  if( image == null ) {

    return null;

  }

  try{

    const {uri} = await FileSystem.getInfoAsync(image);

    const blob = await new Promise((resolve, reject) => {

      const xhr = new XMLHttpRequest();

      xhr.onload = function() {

        resolve(xhr.response);

      }

    })

    ...

  }

}


```

```
};

xhr.onerror = function(e) {

    console.log(e);

    reject(new TypeError('Network request failed'));

};

xhr.responseType = 'blob';

xhr.open('GET', uri, true);

xhr.send(null);

});

setUploading(true);

const filename = image.substring(image.lastIndexOf('/') + 1);

const storageRef = firebase.storage().ref(`photos/${filename}`);

const task = storageRef.put(blob);

return new Promise((resolve, reject) => {

    task.on(

        'state_changed',

        (taskSnapshot) => {

            const progress =
```

```
(taskSnapshot.bytesTransferred / taskSnapshot.totalBytes) * 100;

console.log(`Upload is ${progress.toFixed(0)}% done`);

setTransferred(progress.toFixed(0));

// You can update the UI with the upload progress if needed

},

(error) => {

  console.log('Error uploading image:', error);

  reject(error); // Reject the promise if upload fails

},

async () => {

  // If upload is successful, get the download URL

  const downloadURL = await storageRef.getDownloadURL();

  console.log('Image uploaded successfully. Download URL:', downloadURL);

  setUploading(false);

  setImage(null);

  resolve(downloadURL); // Resolve the promise with the download URL

}

);

});
```

```
    } catch (error) {  
  
      console.log('Error:', error);  
  
    }  
  
};  
  
return (  
  
<View style={styles.container}>  
  
<Text>Select Image:</Text>  
  
<View style={styles.buttonContainer}>  
  
<View style={styles.button3}>  
  
<Button title="Go to previous screen" onPress={() => navigation.navigate("JobPost")}>  
  
</View>  
  
<View style={styles.button1}>  
  
<Button title="Take Photo" onPress={takePhotoFromCamera} />
```

```
</View>

<View style={styles.button2}>
  <Button title="Choose Photo" onPress={choosePhotoFromLibrary} />
</View>

</View>

<InputWrapper>
  {image && <Image source={{ uri: image }} style={{ width: 200, height: 200 }} />}
</InputWrapper>

<InputField
  placeholder="What's on your mind?"
  multiline
  numberOfLines={4}
  value={post}
  onChangeText={(content) => setPost(content)}
/>

{uploading ? (

```

```
<StatusWrapper>

<Text>{transferred} % Completed!</Text>

<ActivityIndicator size="large" color="#0000ff" />

</StatusWrapper>

) : (


<SubmitBtn onPress={submitPost}>

<SubmitBtnText>Post</SubmitBtnText>

</SubmitBtn>

)}


</InputWrapper>

</View>

);

};

export default Addpostscreen;

const styles = StyleSheet.create(


  container: {

    flex: 1,
```

```
    alignItems: 'center',  
    justifyContent: 'center',  
    paddingHorizontal: 20,  
,  
    input: {  
        borderWidth: 1,  
        borderColor: '#ccc',  
        borderRadius: 5,  
        padding: 10,  
        marginVertical: 10,  
        width: '100%',  
        height: 100,  
,  
        progress: {  
            alignItems: 'center',  
            marginTop: 20,  
,  
        },  
    },  
    buttonContainer: {  
        marginTop: 30,  
,  
    },
```

```
button1: {  
  marginBottom: 20,  
  
},  
  
button3:{  
  marginBottom: 20,  
  
},  
});
```

HomePage.js:

```
import {createDrawerNavigator} from '@react-navigation/drawer';  
  
import * as React from "react";  
  
import { Image } from "expo-image";  
  
import { StyleSheet, View, Text, Pressable, Alert, ScrollView } from "react-native";  
  
import { LinearGradient } from "expo-linear-gradient";  
  
import { Color, FontFamily, Border } from "../GlobalStyles1";  
  
import * as Location from 'expo-location';  
  
import * as IntentLauncher from 'expo-intent-launcher';  
  
import { useNavigation } from "@react-navigation/native";  
  
// import { Image } from 'react-native-expo-image-cache';  
450
```

```
const Drawer = createDrawerNavigator();

const HomeScreen = () => {

  const navigation = useNavigation();

  const handlePress = async () => {

    // Check if location services are enabled

    let { status } = await Location.requestForegroundPermissionsAsync();

    if (status !== 'granted') {

      Alert.alert('Location permission not granted', 'Please enable location services to find

nearby hospitals.');

      return;
    }
  }

  // Get current location

  let location = await Location.getCurrentPositionAsync({ });

  const { latitude, longitude } = location.coords;

  // Form Google Maps URL with current location

  const mapsUrl =
`https://www.google.com/maps/search/software+firms+companies/@${latitude},${longitude}`
```

```
,15z`;

// Open Google Maps or fallback to browser

try {

    await IntentLauncher.startActivityAsync('android.intent.action.VIEW', {

        data: mapsUrl,

    });

} catch (error) {

    console.error('Error opening Google Maps:', error);

    Alert.alert('Error', 'Failed to open Google Maps. Please try again later.');

}

};

return (

<ScrollView >

<View style={styles.onboarding}>

<View style={styles.frameParent}>

<Image

style={[styles.groupChild, styles.groupChildPosition]}

contentFit="cover"

source={require("../assets/frame-27.png")}


```

```
/>

<Image

  style={[styles.groupItem, styles.groupLayout]}

  contentFit="cover"

  source={require("../assets/frame-28.png")}

/>

<Image

  style={ styles.groupInner }

  contentFit="cover"

  source={require("../assets/frame-32.png")}

/>

<Image

  style={[styles.frameIcon, styles.frameIconLayout]}

  contentFit="cover"

  source={require("../assets/frame-31.png")}

/>

<Image

  style={[styles.groupControl1, styles.groupControlPosition]}

  contentFit="cover"

  source={require("../assets/frame-33.png")}

/>
```

```
<Image  
    style={styles.groupChild2}  
    contentFit="cover"  
    source={require("../assets/frame-24.png")}  
/>  
  
<Image  
    style={styles.groupChild3}  
    contentFit="cover"  
    source={require("../assets/frame-321.png")}  
/>  
  
<Image  
    style={styles.groupChild4}  
    contentFit="cover"  
    source={require("../assets/frame-34.png")}  
/>  
  
<Image  
    style={styles.groupChild5}  
    contentFit="cover"  
    source={require("../assets/frame-26.png")}  
/>  
  
<Image
```

```
    style={styles.groupChild6}

    contentFit="cover"

    source={require("../assets/frame-311.png")}

/>

<Image

    style={[styles.groupChild7, styles.childPosition]}

    contentFit="cover"

    source={require("../assets/frame-29.png")}

/>

<Image

    style={[styles.groupChild8, styles.groupLayout]}

    contentFit="cover"

    source={require("../assets/frame-331.png")}

/>

<Image

    style={styles.groupChild9}

    contentFit="cover"

    source={require("../assets/frame-312.png")}

/>

<Image

    style={styles.groupChild10}
```

```
contentFit="cover"

source={require("../assets/frame-30.png")}

/>


<Image

style={[styles.groupChild11, styles.frameIconLayout]}

contentFit="cover"

source={require("../assets/frame-322.png")}

/>


<Image

style={styles.groupChild12}

contentFit="cover"

source={require("../assets/frame-25.png")}

/>


<LinearGradient

style={styles.rectangleLineargradient}

locations={[0, 1]}

colors={[`rgba(31, 31, 31, 0)`, "#1f1f1f"]}

/>


</View>

<View style={styles.yourSearchForTheNextDreamParent}>

<Text
```

```
style={[styles.yourSearchForContainer, styles.startSearchingTypo]}

>

<Text

style={styles.yourSearchFor}

>`Your search for the next dream job is over `}</Text>

<Text style={styles.text}>🔗 </Text>

</Text>

<Pressable onPress={handlePress}>

<View style={[styles.startSearchingParent, styles.frameChildFlexBox]}>

<Text style={[styles.startSearching, styles.startSearchingTypo]}>

Nearby Software Companies

</Text>

<* <Button

title="Nearby Hospital"

onPress={handlePress}

/> *}

<Image

style={styles.arrowRightIcon}

contentFit="cover"

source={require("../assets/arrowright.png")}>
```

```
    />

  </View>

</Pressable>

<Pressable onPress={() => navigation.navigate("Chatbot")}>

  <View style={styles.imageContainer}>

    <Image style={styles.image} source= {require("../assets/download32.jpeg")}>/>

  </View>

</Pressable>

<View style={[styles.frameChild, styles.frameChildFlexBox]}>

  </View>

</View>

</ScrollView>

);

};

const styles = StyleSheet.create({
```

```
imageContainer: {  
    marginLeft: 170,  
    marginTop: 40,  
},  
  
image: {  
    width: 150,  
    height: 150,  
    borderRadius: 80,  
    marginBottom: 5,  
},  
  
groupChildPosition: {  
    left: 7,  
    position: "absolute",  
},  
  
groupLayout: {  
    height: 76,  
    borderRadius: 39,  
    position: "absolute",
```

```
},  
  
frameIconLayout: {  
  
    width: 173,  
  
    position: "absolute",  
  
},  
  
childPosition: {  
  
    left: 0,  
  
    position: "absolute",  
  
},  
  
startSearchingTypo: {  
  
    textAlign: "center",  
  
    color: Color.colorWhite,  
  
    //  
  
    //fontFamily: FontFamily.satoshi,  
  
    fontWeight: "700",  
  
},  
  
frameChildFlexBox: {  
  
    justifyContent: "center",  
  
    width: 340,  
  
    borderRadius: Border.br_13xl,  
  
    alignItems: "center",
```

```
    },  
  
    groupChild: {  
  
        top: 379,  
  
        borderRadius: 47,  
  
        width: 185,  
  
        height: 92,  
  
    },  
  
    groupItem: {  
  
        top: 396,  
  
        left: 196,  
  
        width: 175,  
  
    },  
  
    groupInner: {  
  
        top: 316,  
  
        left: 371,  
  
        width: 135,  
  
        height: 165,  
  
        borderRadius: 39,  
  
        position: "absolute",  
  
    },  
  
    frameIcon: {
```

top: 267,

left: 230,

height: 147,

borderRadius: 39,

},

groupChild1: {

top: 298,

width: 248,

height: 90,

borderRadius: 39,

},

groupChild2: {

top: 212,

left: 38,

borderRadius: 44,

width: 189,

height: 86,

position: "absolute",

},

groupChild3: {

top: 205,

left: 227,
width: 169,
height: 87,
borderRadius: 39,
position: "absolute",
},

groupChild4: {
top: 231,
left: 369,
borderRadius: 33,
width: 116,
height: 121,
position: "absolute",
},

groupChild5: {
top: 174,
left: 320,
borderRadius: 26,
width: 138,
height: 61,
position: "absolute",

```
},  
  
groupChild6: {  
  
    top: 149,  
  
    left: 171,  
  
    borderRadius: 32,  
  
    width: 145,  
  
    height: 57,  
  
    position: "absolute",  
  
},  
  
groupChild7: {  
  
    top: 116,  
  
    borderRadius: 43,  
  
    width: 171,  
  
    height: 117,  
  
},  
  
groupChild8: {  
  
    top: 75,  
  
    left: 169,  
  
    width: 239,  
  
},  
  
groupChild9: {
```

```
    top: 34,  
    left: 404,  
    width: 76,  
    height: 148,  
    borderRadius: 39,  
    position: "absolute",  
,  
groupChild10: {  
    top: 6,  
    left: 266,  
    width: 148,  
    height: 78,  
    borderRadius: 39,  
    position: "absolute",  
,  
groupChild11: {  
    top: 0,  
    left: 91,  
    borderRadius: 41,  
    height: 82,  
,
```

```
groupChild12: {  
    top: 78,  
    left: 33,  
    width: 117,  
    height: 62,  
    position: "absolute",  
    borderRadius: 24,  
},  
  
rectangleLineargradient: {  
    top: 246,  
    left: 51,  
    width: 430,  
    height: 243,  
    backgroundColor: "transparent",  
    position: "absolute",  
},  
  
frameParent: {  
    top: 466,  
    left: -51,  
    width: 506,  
    height: 489,
```

```
position: "absolute",  
},  
  
yourSearchFor: {  
  
  display: "flex",  
  
  fontSize: 30,  
  
  marginLeft: 16,  
  
},  
  
text: {  
  
  fontSize: 32,  
  
},  
  
yourSearchForContainer: {  
  
  alignSelf: "stretch",  
  
  zIndex: 0,  
  
},  
  
startSearching: {  
  
  fontSize: 16,  
  
},  
  
arrowRightIcon: {  
  
  width: 24,  
  
  height: 24,  
  
  marginLeft: 8,
```

```
        },  
  
    startSearchingParent: {  
  
        backgroundColor: "#5424fd",  
  
        flexDirection: "row",  
  
        padding: 16,  
  
        zIndex: 1,  
  
        marginTop: 40,  
  
    },  
  
    frameChild: {  
  
        top: 235,  
  
        backgroundColor: "rgba(84, 36, 253, 0.6)",  
  
        zIndex: 2,  
  
        left: 0,  
  
        position: "absolute",  
  
    },  
  
    yourSearchForTheNextDreamParent: {  
  
        top: 79,  
  
        left: 45,  
  
        alignItems: "center",  
  
        position: "absolute",  
  
    },
```

```
onboarding: {  
  backgroundColor: "#1f1f1f",  
  
  flex: 1,  
  
  width: "100%",  
  
  height: 932,  
  
  overflow: "hidden",  
  
  borderRadius: 0,  
  
},  
});  
  
export default HomeScreen;
```

5.21 Test result and report:

	Features	To be tested
Student	Students creates an account	YES
	Update Account	YES
	Generates Personal portfolio	YES
	View Job Dues	YES
	View Meal Chart	NO
	View House Tutor	NO
	Give Complaints	YES
	Give Feedback	NO
Admin	Approve Student	YES
	Update Job Dues	YES
	Post News	NO
	Update Student Profile	YES
	View Complaints	NO
	View Feedback	NO
	Update Settings	YES
	Upload House Tutor	YES

Table 5.21.1: Testing Result and report

Test Case No	Test Description	Testing Steps	Test Data	Expected Result
TC 01	Validate input details in the login form	1. Enter the user's name according to test data 2. Enter the password according to test data 3. Press login button	Blank user name & blank password	Error message "Both Email and password are Empty"
TC 02			Blank user name & given a password	Error message "Empty Email"
TC 03			Given user name & blank Password	Error message "Empty password"
TC 04			Wrong email and correct password	Error message "Invalid Email address or Password"
TC 05			Correct email and wrong password	Error message "Invalid Email address or Password"
TC 06			Both incorrect email & password	Error message "Invalid Email address or Password"
TC 07			Correct email & correct password	Redirect to Dashboard

Table 5.21.2: Login test case

Test Case No	Test Description	Testing Steps	Test Data	Expected Result
TC 01	Validate inputdata in add a member form	1.Enter first name, last name, date of birth, gender, ID No, A/L batch, email, telephone no, role, image according to test data	First name empty	Error message “Empty First Name”
TC 02		2. Press the save button	Last name empty	Error message “Empty Last Name”
TC 03			Gender not selected	Error message “Please Select your gender”
TC 04			Date and NIC not matching	Error message “Date and NIC not matching”
TC 05			Empty A/L batch	Error message “Empty A/L batch”
TC 06			Empty email	Error message “Empty email”
TC 07			Email address used before	Error message “Already taken”

Table 5.21.3: Member module test cases

Test Case No	Test Description	Testing Steps	Test Data	Expected Result
TC 01	Validate input data in add a product form	1.Enter the product name, product type, category, size, and price according to the test data 2. Press the save button	Product name empty	Error message “Empty Product Name”
TC 02			Product type empty	Error message “Empty Product Type”
TC 03			Category empty	Error message “Empty Category”
TC 04			Price empty	Error message “Empty Price”

Table 5.2.4: Product module test case

Chapter 6

Impact on Society, Environment and Sustainability

6.1 Impact on Society

The ProConnectEase Job Portal App has a significant impact on society, bringing about various positive changes and benefits. The system revolutionizes the job management process by leveraging technology and automation, thereby enhancing efficiency, transparency, and accessibility. Here, we explore the impact of the system on different aspects of society.

- i. Improved Access to job: The ProConnectEase Job Portal App removes geographical barriers and allows applicants from remote areas to easily apply for job .They would have to travel long distances, spend money on transportation, and face the challenges of accommodation just to submit their CV forms. This often discouraged many talented employee from pursuing higher income job.
- ii. Enhanced Efficiency and Time Savings: With the ProConnectEase Job Portal App, the manual handling of application forms and documents is eliminated. The system automates various tasks such as data entry, document verification, and application tracking, saving significant time and effort. This allows the institution to process applications faster and provide timely updates to applicants.
- iii. Reduction in Errors and Mismanagement: Manual job management processes are prone to errors, misplacement of documents, and data inconsistencies. The online system minimizes such risks by enforcing data validation rules, ensuring accurate and complete information. It also reduces the chances of misplaced or lost documents, as all documents are securely stored in a centralized database. The reduction in errors and mismanagement is one of the key advantages of implementing an online system compared to manual processes.
- iv. Cost and Resource Optimization: The system reduces the need for physical infrastructure, printing of application forms, and manual handling of documents. This leads to cost savings for the institution and helps optimize the utilization of resources. It reduces the need for staff to manually process applications, allowing them to focus on other important tasks. Cost and resource optimization are significant benefits of implementing an online system compared to manual processes.

The ProConnectEase Job Portal App has a profound impact on society. It promotes inclusivity, efficiency, transparency, and sustainability in the education sector. The system's positive impact on society extends beyond the admission process, contributing to the overall development and advancement of education in the region.

6.2 Impact on Environment

The ProConnectEase Job Portal App has a positive impact on the environment in several ways. By digitizing and automating the job management process, the system significantly reduces the consumption of paper and other physical resources. This reduction in paper usage directly contributes to the conservation of trees and reduces the environmental impact associated with paper production, including deforestation, water consumption, and energy consumption.

- v. Reduced Energy Consumption: The system operates on electronic devices and servers, which consume significantly less energy compared to the physical infrastructure required for a traditional apply process. By minimizing energy consumption, the system helps to lower carbon emissions and reduce the overall environmental footprint.
- vi. Conservation of Water Resources: The production of paper involves substantial water usage, from the cultivation of trees to the paper manufacturing process. By reducing the demand for paper through digitization, the system indirectly conserves water resources, as less water is needed for tree growth and paper production. This helps to alleviate water scarcity and minimize the environmental impact on aquatic ecosystems.
- vii. Waste Reduction: The traditional apply process generates a considerable amount of waste in the form of paper documents, prospectuses, and application forms. By eliminating the need for physical documents and embracing a paperless system, the online admission management system helps to reduce waste generation. This reduces the strain on landfill space and minimizes the need for waste disposal, contributing to a cleaner and more sustainable environment.
- viii. Lower Carbon Emissions: The online system reduces the need for physical transportation and commuting by enabling applicants to complete the admission process remotely. This results in reduced carbon emissions from vehicles, as fewer trips are required to visit the institution. The system promotes a greener approach to the admission process and contributes to mitigating climate change by minimizing greenhouse gas emissions.

6.3 Ethical Aspects

The implementation of the ProConnectEase Job Portal App brings about various ethical aspects that should be considered. These include:

- ix. Privacy and Data Protection: The system deals with sensitive personal information of applicants, such as their names, contact details, job records, and financial information. It is crucial to ensure the security and confidentiality of this data, adhering to relevant data protection laws and regulations.
- x. Fairness and Equal Opportunities: The online job system should be designed and implemented in a way that ensures fairness and equal opportunities for all applicants
- xi. Accessibility and Inclusivity: The system should be accessible to all individuals, including those with disabilities or special needs. Ensuring inclusivity allows equal participation and opportunity for all applicants.
- xii. Accuracy and Integrity of Information: The system should maintain the accuracy and integrity of the information provided by applicants. It is essential to implement measures to prevent unauthorized access, tampering, or manipulation of data.
- xiii. Ethical Use of Technology: The use of technology in the job system should align with ethical considerations. It is crucial to respect intellectual property rights, avoid plagiarism or unauthorized use of copyrighted materials, and ensure compliance with software licenses and legal obligations. Ethical practices in technology use foster a culture of integrity and respect for intellectual property.
- xiv. Accountability and Responsibility: Those involved in the management and operation of the online application system should demonstrate accountability and take responsibility for their actions. They should adhere to ethical standards, exercise good judgment, and handle applicant information and resources with care. Transparency, openness to feedback, and addressing concerns or grievances in a responsible manner are crucial aspects of maintaining ethical practices.

6.4 Sustainability Plan

The sustainability plan for the ProConnectEase Job Portal App focuses on implementing strategies and practices to ensure the long-term sustainability, efficiency, and environmental responsibility of the system. Here are the key components of the sustainability plan:

- xv. Energy Efficiency: Optimize the energy consumption of the system by using energy-efficient hardware, servers, and infrastructure. This includes utilizing power-saving features, implementing server virtualization to maximize resource utilization, and regularly monitoring energy consumption to identify areas for improvement.
- xvi. Cloud Infrastructure: Consider utilizing cloud-based infrastructure for hosting the system. Cloud services often provide scalable resources and efficient utilization of computing power, resulting in reduced energy consumption and carbon footprint compared to traditional on-premises infrastructure.
- xvii. Renewable Energy Sources: Explore the possibility of using renewable energy sources to power the system's infrastructure. Consider partnering with renewable energy providers or investing in renewable energy solutions to reduce reliance on fossil fuels and minimize greenhouse gas emissions.
- xviii. Continuous Improvement: Regularly assess the system's performance, efficiency, and sustainability practices. Identify areas for improvement and implement updates or enhancements accordingly.
- xix. Monitoring and Reporting: Establish monitoring mechanisms to track the system's environmental impact, energy consumption, and sustainability initiatives. Regularly report on key metrics and progress towards sustainability goals to demonstrate the institution's commitment to sustainability and accountability.
- xx. Policy Integration: Integrate sustainability considerations and guidelines into the institution's overall policies and practices. Ensure that sustainability is considered in decision-making processes related to the admission management system and that it aligns with the institution's broader sustainability goals and initiatives.

By implementing this sustainability plan, the ProConnectEase Job Portal App aims to minimize its environmental footprint, promote responsible resource usage, and contribute to a more sustainable future. The plan ensures that the system is designed, operated, and maintained with sustainability principles in mind.

Chapter 7

At a Glance Whole project

7.1 At a glance whole project:

Project Name: ProConnectEase

Objective: Develop a comprehensive mobile app solution to connect job seekers with employers and facilitate the recruitment process. It provides a centralized database where job seekers can create profiles, upload resumes, and search for job listings based on various criteria such as job title, location, industry, and experience level. It can also get help from custom chatbot to know anything related to the job for the job seekers and It can also help to create Video CV.

Durations: 12 months **Key Components**

Job Management:

- i. User login and signing through email verification
- ii. Search real time jobs through an API named Jsearch.

Storage Management:

- i. Firebase for handling and saving the data securely
- ii. Used API for cloud-based data

User Management:

- i. Maintain user profiles (name, email).
- ii. Assign roles and responsibilities to users.

Service Management:

- i. Video CV
- ii. Customized Chatbot
- iii. Firebase Validation
- iv. Apply
- v. Search using API
- vi. Firebase Authentication
- vii. User Post Management using Firebase

Notifications:

- i. Send notifications to recruiter(emails and attachments).

Initial Planning(1-2 Months):

- i. Define project scope, objectives, and requirements. Create a detailed project plan and allocate resources.
- ii. Formulate the project team and establish communication.
- iii. Design system architecture, database, and user interfaces.

System Design & Development(3-4 Months):

- i. Develop core functionalities: email authentication, custom profile build.
- ii. Begin integration testing of modules

Feature Development & Testing (5-6 Months):

- i. Develop additional features: video cv, apply, job search, customized chatbot, job posting, resume builder, job delete, communicate with recruiter through email.
- ii. Conduct testing of integrated features.

User Interface Refinement & Testing(7-8 Months):

- i. Address bugs and issues identified during testing. Enhance user interfaces based on feedback.
- ii. Perform end-to-end testing of the system.

Documentation and Training(9-10 Months):

- i. Implement security measures. Create user documentation and training materials.
- ii. Deploy the system ProConnectEase in the Expo Go environment. .

Deployment And Finalization(11-12 Months):

- i. Migrate data and perform final testing.
- ii. Launch the ProConnectEase System in Expo Go environment
- iii. performance and gather user feedback

Key Principal:

- i. Communication
- ii. Collaboration
- iii. Adaptability
- iv. Quality
- v. Risk Management

This overview provides a high-level glance at the project. Detailed planning, coordination, and execution across each phase will be necessary for the successful development and deployment of the ProConnectEase System.

Chapter 8: CONCLUSION & FUTURE SCOPE

8.1 Discussion and Conclusion

This job portal app represents a significant advancement in the recruitment and job search domain by leveraging modern technologies and user-centric features. By incorporating email authentication, custom profile building, video CVs, efficient job search and application processes, interactive chatbot support, and robust communication tools, your app offers a holistic solution for job seekers and employers alike.

The emphasis on user experience, security, and functionality aligns with industry standards and user expectations, making your app a valuable asset in the competitive job market landscape. Moving forward, continuous updates, user feedback incorporation, and staying abreast of emerging trends will further solidify your app's position as a go-to platform for job seekers and recruiters, fostering meaningful connections and successful placements.

8.2 Scope for Future Developments

The following are the future scope for the project.

- i. Should be added one to one communication system with the job seeker and recruiter
- ii. Build a strong API to successfully manage unlimited real time job data
- iii. Increase security system in the app
- iv. Implement question set feature for evaluating job seekers

REFERENCES

- [1] Rajegopal, S., McGuin, P., Waller, J.: Project Portfolio Management: Earning An Execution Premium. Palgrave Macmillan (2007)
- [2] Beverly Pasian Dr Nigel Williams, D., Young, M., Young, R., Romero Zapata, J.: Project, programme and portfolio maturity: a case study of Australian Federal Government. *Int. J. Manag. Proj. Bus.* 7, 215–230 (2014).
- [3] H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling and Controlling, Eight Edition, John Wiley & Sons, 2003.
- [4] I. Lovrek, M. Kos, B. Mikac, "Collaboration between Academia and Industry: Telecommunications and Informatics at the University of Zagreb," *Computer Communications*, vol. 26, pp. 451-459, Elsevier, 2003
- [5] Mieritz, L., Fitzgerald, D., Gomolski, B., Light, M.: Toolkit Best Practices: Program and Portfolio Management Maturity Model. (2007), PP- 2-4.
- [6] Dooley, L., Lupton, G., O'Sullivan, D.: Multiple project management: A modern competitive necessity. *J. Manuf. Technol. Manag.* 16, 466–482 (2005).
- [7] M. Á. R. O. Ana Isabel Molina, "A methodological approach for user interface development of collaborative applications: A case study," *Science Direct*, vol. 74, no. 9, pp. 754-776, 2009.
- [8] T. Jewett, "Models and languages," [Online]. Available: <http://www.tomjewett.com/dbdesign/dbdesign.php?page=models.html>. [Accessed 19 April 2017]. PP 2-5.
- [9] H. W. . M. R. Alan Dennis, *Systems Analysis and Design*, 2008. PP 10-23
- [10] Cooper, R., Edgett, S., Kleinschmidt, E.: Portfolio management for new product development: Results of an industry practices study. *R D Manag.* 31, 361–380 (2001).
- [11] Young, M., Conboy, K.: Contemporary project portfolio management: Reflections on the development of an Australian Competency Standard for Project Portfolio Management. *Int. J. Proj. Manag.* 31, 1089–1100 (2013).
- [12] E. Erno-Kjolhede, "Project Management Theory and the Management of Research Projects", MPP Working Paper No. 3/2000, ISBN:87-90403-70-3, Denmark, 2000.

- [13] Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: The design science research process: a model for producing and presenting information systems research. In: Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006). pp. 83–106 (2006).
- [14] Mikkola, J.H.: Portfolio management of R&D projects: implications for innovation management. *Technovation*. 21, 423–435 (2001).
- [15] Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of information systems. *MIS Q.* 369–386 (1987).
- [16] Lima, A., Monteiro, P., Fernandes, G., Machado, R.J.: Dependency Analysis between PMI Portfolio Management Processes. In: ICCSA'2016, Lecture Notes in Computer Science, Springer, 2016. (accepted for publication), Beijing, China (2016).
- [17] Young, M., Conboy, K.: Contemporary project portfolio management: Reflections on the development of an Australian Competency Standard for Project Portfolio Management. *Int. J. Proj. Manag.* 31, 1089–1100 (2013).