

# AntHill

A Project by Group 3  
Final Demo

What is the Problem AntHill Addresses?

# Premium Account

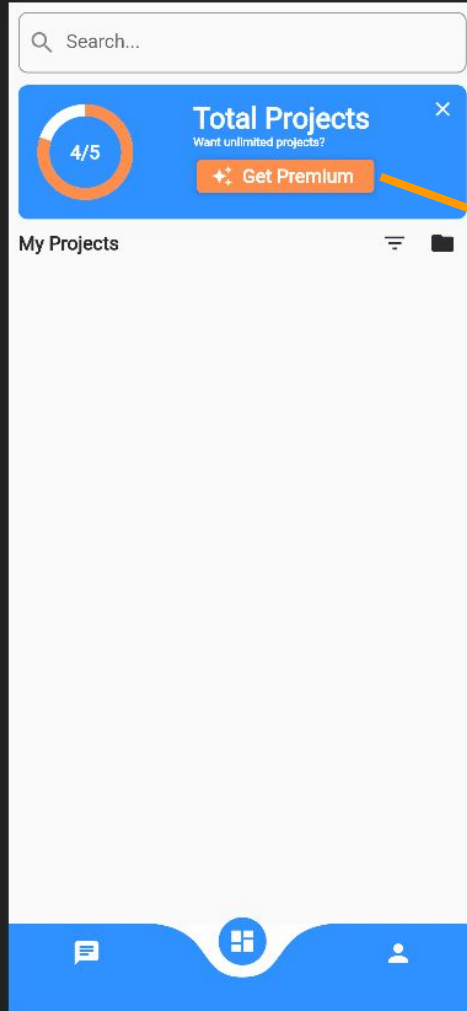
Anirudha Kyathsandra

## Need for a premium account

- Need someone to create projects
- Manage projects
- Meant for professors, departments, companies

# Privileges of a Premium User

- Ability to create projects
- Ability to take on projects as well
- Chat with potential students and clients via a chat feature



## Select Account

### Free Account \$0/month

- Take on up to 3 projects at a time
  - Get JIRA access on your project dashboards
  - Communicate with clients via a chat feature
- All this for FREE!

🏠 Use Free Plan

### Get Premium \$9.99/month

Get the premium version of Anthill and get your work done by posting projects that can be taken on by the talented students on this App. Dont worry you can still take on projects as well!

- Take on up to 5 projects at a time
- Create and post projects for other students
- Communicate with clients via a chat feature

All this just for \$9.99!

➤ Get Premium

Return Home

```
Padding(  
  padding: const EdgeInsets.all(8),  
  child: ElevatedButton.icon(  
    onPressed: () {  
      Navigator.pop(context);  
      Navigator.push(context,  
        MaterialPageRoute(builder: (context) {  
          return const PaymentScreen();  
        })); // MaterialPageRoute  
    },  
    icon: const Icon(Icons.auto_awesome, size: 18),  
    label: Text('Get Premium',
```

Get a premium account

← Confirm Payment

Card Holders Name

John Doe

Credit Card Number

Expiration

CVV

Zip Code

Payment page that stores a hashmap of string- user input.

```
final Map<String, Map<String, dynamic>> textFieldContents = {  
  'Card Holders Name': {  
    'initialValue': '',  
    'hint': 'John Doe',  
    'icon': const Icon(Icons.person)  
  },  
  'Credit Card Number': {  
    'initialValue': '',  
    'hint': '0000 0000 0000 0000',  
    'icon': const Icon(Icons.credit_card)  
  },  
}
```

← Confirm Payment

Card Holders Name

John Doe

Credit Card Number

0000 0000 0000 0000

Expiration

12/24

CVV

000

Zip Code

00000

← **Confirm Payment**

Card Holders Name  
John Doe

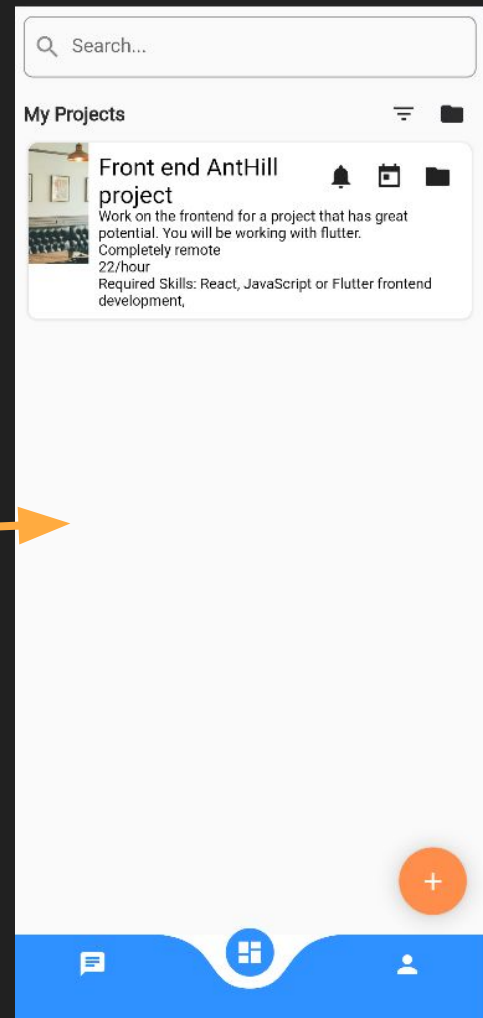
Credit Card Number  
0000 0000 0000 0000

Expiration  
12/24

CVV  
000

Zip Code  
00000

Payment page that sets the premium boolean to true and takes you to the premium dashboard screen.





# Challenges

1. Deciding how to have users create projects, who to give project create permissions to?
2. How to distinguish a user as a premium user vs a normal user?
3. What fields to add for a create project?
4. Making the switch between the dashboard for the regular user and premium user.

# Solutions

1. Deciding to have a premium account and have an add a project feature.
2. Setting boolean values based on the payment page. Once a user hits get premium button, and payment goes through, set boolean to true.
3. Decided to add a project name, description field to keep things simple for now. Future implementations would include company, pay, etc.
4. Worked with Ethan to update the page to incorporate the create project button and remove the get premium feature.

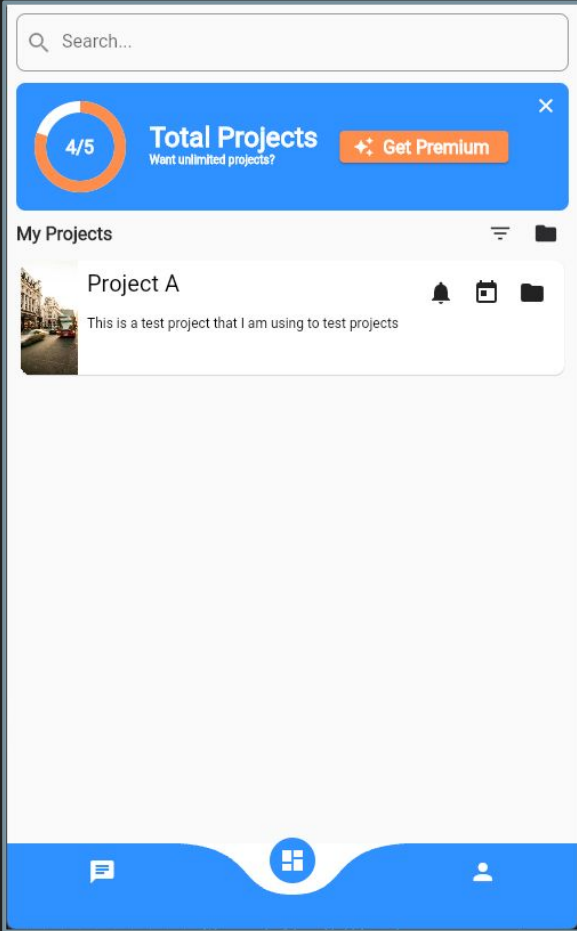
# Personal Accomplishments

- Coming up with the app idea and following through with why this was the best project idea
- Designing the original UI
- Learning flutter, how to use it, configure it and code in it
- Continuous design changes -scrum/agile
- Adding features to the project details page and project creation page
- Implementing the base profile page
- Implementing the whole premium page section
- Following and learning good software development practices
- Working on the design documents



# Profile, Dashboard, & Refactoring

Ethan Lindsey

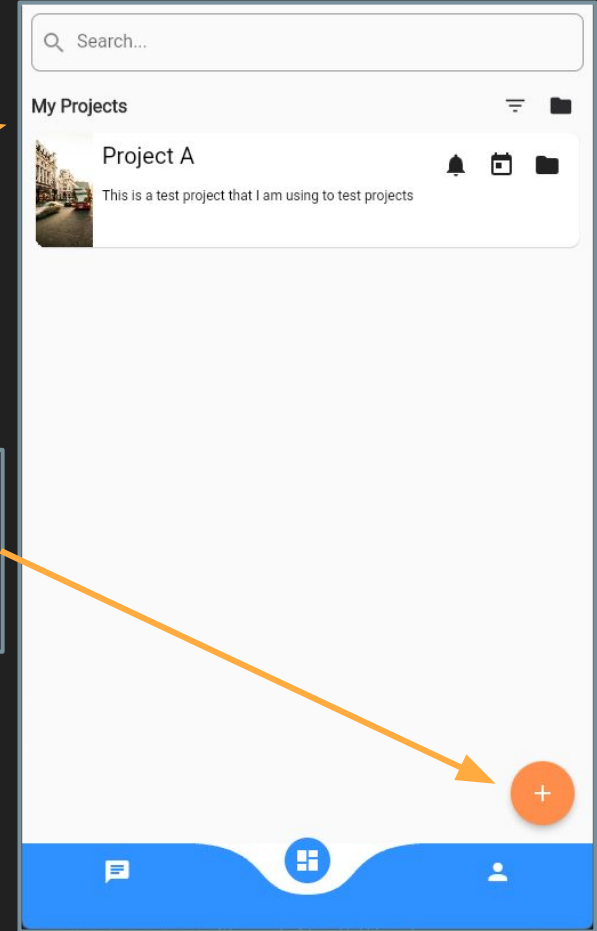


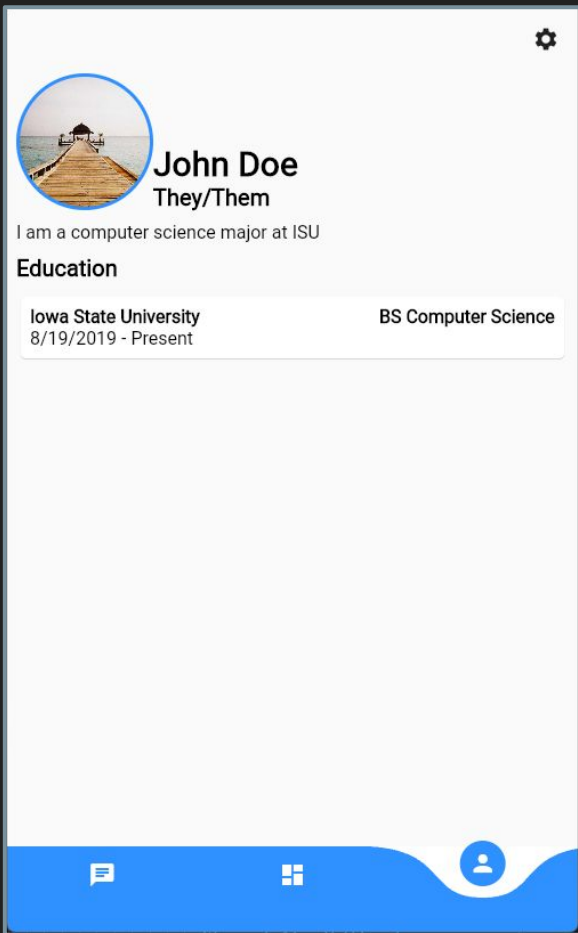
No Ad For Premium Users

```
//Premium Bar  
if (!_premium && !_promptPremium) _getPremiumBanner(context),
```

```
floatingActionButton: !_premium  
? null  
: FloatingActionButton(  
  onPressed: () => Navigator.push(  
    context,  
    MaterialPageRoute(  
      builder: (context) => const CreateProjectForm()),  
      child: const Icon(Icons.add)), // FloatingActionButton
```

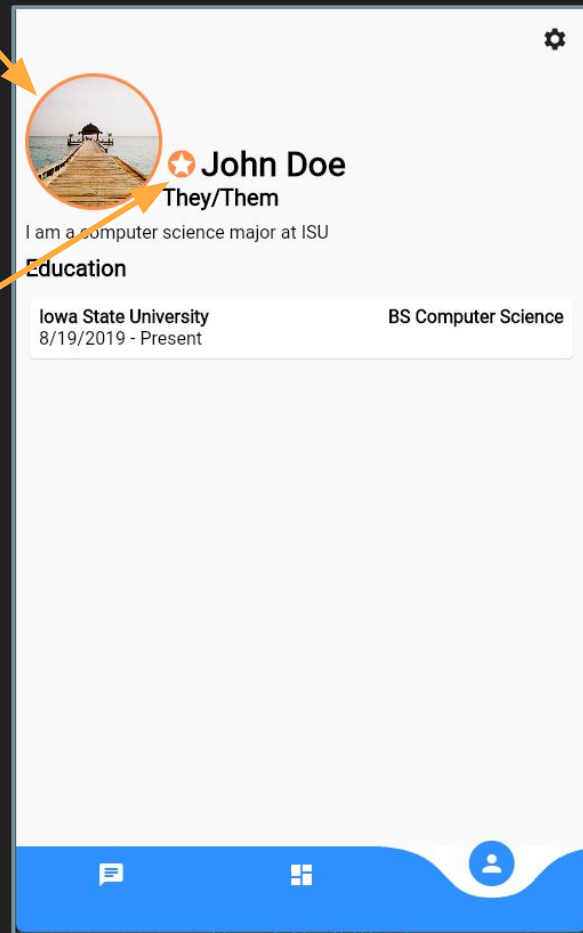
Create Project Button for Premium





```
color: _data['premium'] ? color.secondary : color.primary,
```

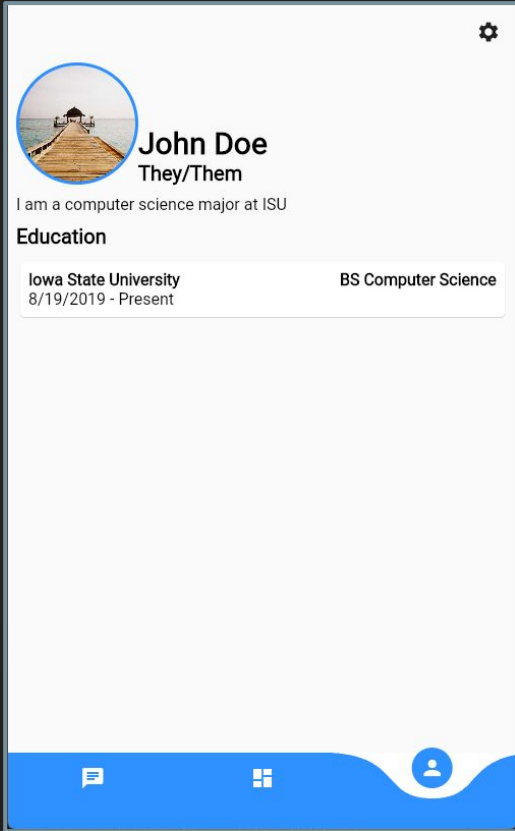
# Conditionally Sets Color



```
if (_data['premium']) ...[  
  const SizedBox(width: 5),  
  Container(  
    decoration: BoxDecoration(  
      color: color.secondary, shape: BoxShape.circle),  
    child: Icon(Icons.star, color: color.background), /  
    const SizedBox(width: 5)  
  ],
```

# Displays Premium Badge

# Before Edit



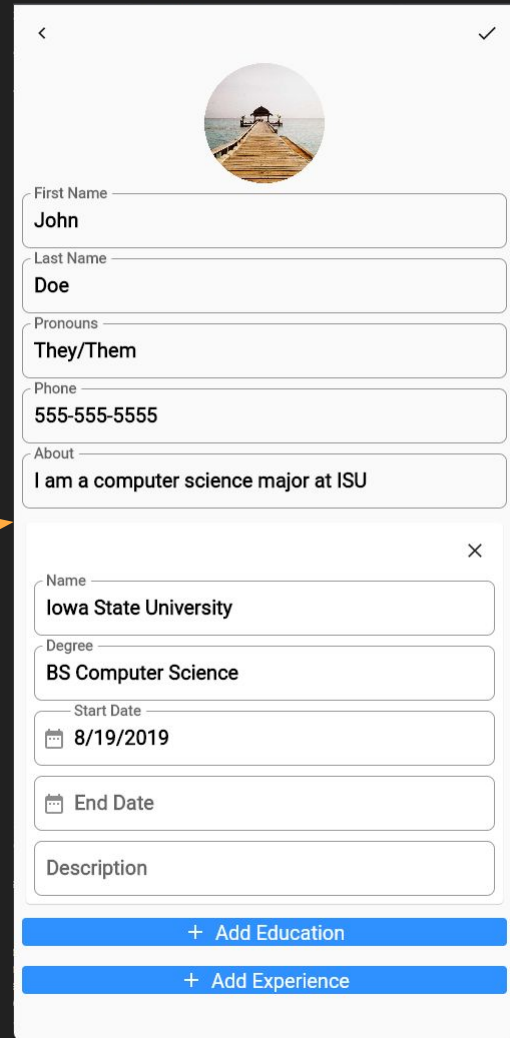
A user profile card for John Doe. At the top left is a circular profile picture of a wooden pier extending into the ocean. To the right of the picture, the name "John Doe" is displayed in a large font, with "They/Them" below it in a smaller font. Underneath the name is the bio: "I am a computer science major at ISU". Below the bio is a section titled "Education" which contains a single entry: "Iowa State University" with the dates "8/19/2019 - Present" and "BS Computer Science" to its right. A gear icon is in the top right corner. At the bottom is a blue navigation bar with icons for messages, a grid, and a profile.

**John Doe**  
They/Them

I am a computer science major at ISU

**Education**

Iowa State University	8/19/2019 - Present	BS Computer Science
-----------------------	---------------------	---------------------



An edit profile form with a back arrow in the top left and a checkmark in the top right. It features a circular profile picture at the top. Below it are input fields for "First Name" (John), "Last Name" (Doe), "Pronouns" (They/Them), and "Phone" (555-555-5555). An "About" section contains the text "I am a computer science major at ISU". Below this is a section for "Education" with a close button (X) in the top right. It contains fields for "Name" (Iowa State University), "Degree" (BS Computer Science), "Start Date" (8/19/2019), and "End Date". A "Description" field is also present. At the bottom are two blue buttons: "+ Add Education" and "+ Add Experience".

First Name  
**John**

Last Name  
**Doe**

Pronouns  
**They/Them**

Phone  
**555-555-5555**

About  
**I am a computer science major at ISU**

Name  
**Iowa State University**

Degree  
**BS Computer Science**

Start Date  
**8/19/2019**

End Date


Description

+ Add Education

+ Add Experience

# During Edit

< ✓



First Name  
**John**

Last Name  
**Doe**

Pronouns  
**They/Them**

Phone  
**555-555-5555**

About  
**I am a computer science major at ISU**

×

Name  
**Iowa State University**

Degree  
**BS Computer Science**

Start Date  
**8/19/2019**

End Date


Description

+ Add Education

+ Add Experience



< ✓



First Name  
**Joey**

Last Name  
**Donut**

Pronouns  
**They/Them**

Phone  
**555-555-5555**

About  
**I am a computer science major at ISU**

×

Name  
**Iowa State University**

Degree  
**BS Computer Science**

Start Date  
**8/19/2019**

End Date

Description

+ Add Education

×

Name  
**John Deere**


Title  
**Software Engineer**

Start Date  
**12/1/2022**

End Date

+ Add Experience

< ✓



First Name  
**Joey**

Last Name  
**Donut**

Pronouns  
**They/Them**

Phone  
**555-555-5555**

About  
**I am a computer science major at ISU**

Name ×  
**Iowa State University**

Degree  
**BS Computer Science**

Start Date  
8/19/2019

End Date

Description

+ Add Education

Name ×  
**John Deere**

Title  
**Software Engineer**


Start Date  
12/1/2022

End Date

+ Add Experience

## After Edit

⚙️



**Joey Donut**  
They/Them

I am a computer science major at ISU

**Education**

<b>Iowa State University</b> 8/19/2019 - Present	<b>BS Computer Science</b>
---	----------------------------

**Experience**

<b>John Deere</b> 12/1/2022 - Present	<b>Software Engineer</b>
--	--------------------------

💬 🏠 👤



# User

```
JCgJdzYjt2PNIArwqs4jxZIIDRw1
+ Start collection
+ Add field
  about: "I am a computer science major at ISU"
  city: "Ames"
  country: "US"
  education
    0
      degree: "BS Computer Science"
      description: null
      endDate: null
      name: "Iowa State University"
      startDate: August 19, 2019 at 12:00:00 AM UTC-5
      email: "test@email.com"
  experience
    firstname: "John"
    lastname: "Doe"
    linkedin: "NA"
    phone: "555-555-5555"
    premium: true
  projects
    0 "jGq6urNL4gYelomDvi2m"
    1 "gurCZ0iY9PZqefqehUFx"
    2 "KvG85gvAD9wUoYXsEcSO"
```

# Project A

```
jGq6urNL4gYelomDvi2m
+ Start collection
+ Add field
  about: "This is a test project that I am using to test projects"
  name: "Project A"
```

# Project B

```
gurCZ0iY9PZqefqehUFx
+ Start collection
+ Add field
  about: "Hey! Come work on my project! we're doing AI for checkers!"
  name: "Project B"
```

# Project C

```
KvG85gvAD9wUoYXsEcSO
+ Start collection
+ Add field
  about: "Hey! Have you seen those checkers people? I heard their AI is written in Cobalt. Come work for us and use Python!"
  name: "Project C"
```

```
StreamBuilder<QuerySnapshot<Map<String, ...
  Column
    ProjectCard
    ProjectCard
    ProjectCard
```

Search...

My Projects

- Project C  
Hey! Have you seen those checkers people? I heard their AI is written in Cobalt. Come work for us and use Python!
- Project B  
Hey! Come work on my project! we're doing AI for checkers!
- Project A  
This is a test project that I am using to test projects

# Project Card

✓  lib/screens/dashboard/dashboard\_screen.dart 

+169 -757



View file @e2ee376a

>  lib/widgets/project\_card.dart  0 → 100644

+123 -0



View file @e2ee376a

~61% Reduction

(Only for Project Card)

# Other Encapsulations

All items were originally repeated in their pages, making code unmaintainable and inflated

- Education Card
- Education Edit Card
- Experience Card
- Experience Edit Card
- Chat Card
- Settings Card

# Challenges

1. Profile Edit screen being ahead of the Registration screen caused errors.
2. Education and Experience is dynamic in length and user specific.
3. Refactoring required reading and editing a lot of repeated code which would take a lot of time.
4. The nested code style of Flutter became cluttered with many layers.

# Solutions

1. Both Profile screen and Profile Edit screen have null checking.
2. Stored Education and Experience as array in user data instead of separate collection.
3. Learned shortcuts and tools for multiline editing in Visual Studio Code.
4. Encapsulate certain components into functions and classes to reduce direct nesting.

# Personal Accomplishments

- Became comfortable with Dart, Flutter, and Firebase
- Implemented central features (Dashboard & Profile Editing)
- Improved both the efficiency and security of existing code
- Reduced file size and increased readability with encapsulation
- Designed the logo



# Chat Screen

Ishan Banerjee

About

---

Country

---

Zip Code

---

City

---

☑ Phone Number

---

☑ LinkedIn

+ Add Education

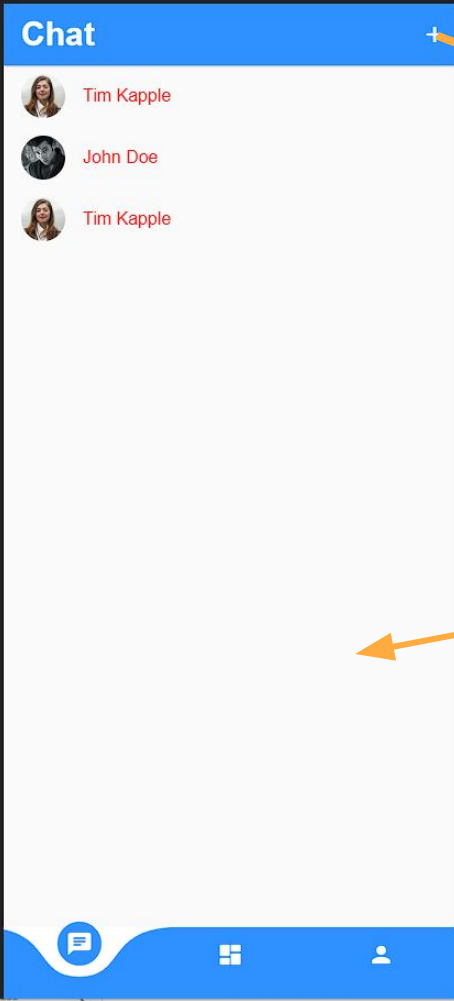
+ Add Experience

Next

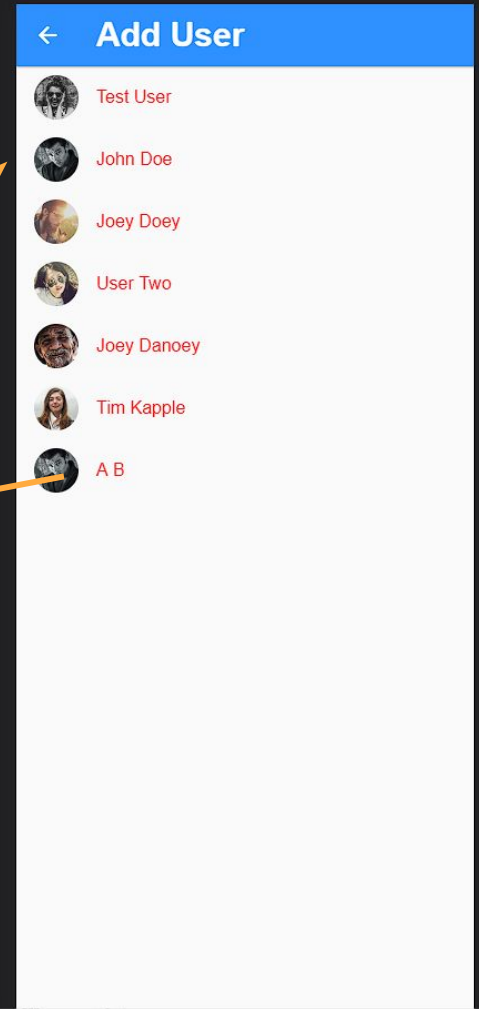
The screenshot shows the Firestore console with three panels:

- Left Panel:** Collections list including 'accounts', 'projects', 'rooms', and 'users' (highlighted with an orange arrow).
- Middle Panel:** Document list for the 'users' collection, showing several document IDs and a document ID 'zkzKMc27iMR8BESaAPwIgsBCFUj2' (highlighted with an orange arrow).
- Right Panel:** Document details for 'zkzKMc27iMR8BESaAPwIgsBCFUj2', showing fields like 'createdAt', 'firstName', 'imageUrl', 'lastName', 'lastSeen', 'metadata', 'role', and 'updatedAt'.

```
await FirebaseChatCore.instance.createUserInFirestore(types.User(  
    firstName: firstName,  
    id: cred.user!.uid,  
    imageUrl: 'https://i.pravatar.cc/300?u=$email',  
    lastName: lastName));  
} on FirebaseAuthException catch (e) {  
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
        content: Text(e.message ?? ''),  
        backgroundColor: Theme.of(ctx).errorColor)); // SnackBar  
}
```



```
IconButton(  
  onPressed: _user == null  
    ? null  
    : () {  
      Navigator.of(context).push(  
        MaterialPageRoute(  
          // Route to go to the chat adding screen  
          builder: (context) => const ChatUserAdd(),  
        ), // MaterialPageRoute  
      );  
    },  
  icon: const Icon(Icons.add) // IconButton  
),  
...
```





13:10

Hello

Hey

this is a test message from Ishan

+ Message

+ Start collection	+ Add document	+ Start collection
accounts	11SemTiq6GFmsaBmLBB9	+ Add field
projects	1nsSea1eX9u4qvtZU1wq	createdAt: 14 December 2022 at 13:47:49 UTC-6
rooms	67Ka55iW0mIq2Akh9Br3	imageUrl: null
users	BhLSIdH2LqeB83ruAZzk	metadata: null
	EBGW6potMvAUzTdodn93	name: null
	WpxSZhmbrrEekiv584on	type: 'direct'
	ccQx0U6jIKIwweR8m86i	updatedAt: 14 December 2022 at 13:47:49 UTC-6
	dXYbd7SnQrjLzUmHg1Hq	userIds
	gM6nqaAJzXHCero0G1n9	0 "8kM0WsJbpPZidMmMG09IAi6119t2"
	hI6N8I3wVEBRACC961sI	1 "JmIucsY0tRXA6kz7aNaCr5Y8NGe2"
	uockTBIwWTabwhU3Qq7y	userRoles: null

13:10

Hello

Hey

this is a test message from Ishan

+ Message

```
/// Handles sending the message
void _handleSendPressed(types.PartialText message) {
  FirebaseChatCore.instance.sendMessage(message, widget.room.id);
}
```

# Challenges

1. Choosing the correct approach to the chatting system was perplexing.
2. Implementing point-point communication system was complex.
3. Connecting the GUI to the backend was tricky
4. Integrating the chat system with the app

# Solutions

1. Trying out various libraries and reading documentations for chatting systems
2. Use FlutterFireChatCore gave the functionality to perform point to point communication.
3. Using a flutters streambuilder to create data streams to connect the GUI with backend.
4. Teamwork and communication helped to integrate the chat feature with the app.

# Personal Accomplishments

- Learned Dart, Flutter
- Firebase integration and features
- Learned the chat communication systems and implementation
- Learned about database query optimization to make retrieving data faster
- Get the chat system running



# Registration, Project, & Database

Nate Reinboldt

# Sign In Screen

Don't have an account? [Sign Up](#)

```
TextSpan(  
  text: 'Sign Up',  
  recognizer: TapGestureRecognizer()  
    ..onTap = () => widget.toggleView(),  
  style: TextStyle(  
    color: color.primary, fontWeight: FontWeight.bold))
```

# Sign Up Screen

Already have an account? [Sign In](#)

```
TextSpan(  
  text: 'Sign In',  
  recognizer: TapGestureRecognizer()  
    ..onTap = () => widget.toggleView(),  
  style: TextStyle(color: color.primary, fontWeight: FontWeight.bold))
```

# Authentication Screen

```
void toggleView() {  
  setState(() => showSignIn = !showSignIn);  
}
```

```
@override  
Widget build(BuildContext context) {  
  if (showSignIn) {  
    return SignIn(toggleView: toggleView, submitFn: _submitLoginForm);  
  } else {  
    return SignUp(toggleView: toggleView, submitForm: _submitForm);  
  }  
}
```

# Sign Up Screen

Form fields for sign up:

- About
- Country
- Zip Code
- City
- Phone Number
- LinkedIn

Buttons:

- + Add Education
- + Add Experience
- Next

# FireBase Database

```
about: "Computer science student looking for coding experience"  
city: "Saint Paul" (string)  
country: "USA"  
education  
  0  
    degree: "Computer Science"  
    description: "Bachelors degree in Computer Science"  
    name: "Iowa State University"  
    startDate: August 20, 2018 at 12:00:00 AM UTC-5  
email: "demo@email.com"  
experience  
  0
```

# Profile Screen

Profile information for Test1 User:

- Profile Picture:
- Username: Test1 User
- Role: Other
- Greeting: Hello, this is a demo
- Education:
  - Iowa State (12/20/2018 - 12/14/2022) - Computer Science
- Experience:
  - Job One (12/6/2022 - 12/14/2022) - Test
  - Job Two (12/1/2022 - Present) - Test
  - Job Three (12/6/2022 - 12/14/2022) - Test



# Create Project Page

Project Name

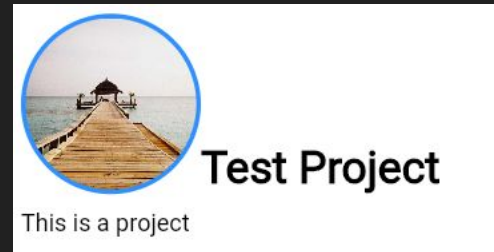
About

Create Project

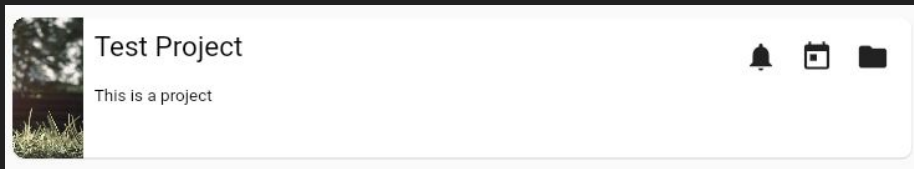
## FireBase Database

```
about : "This is a project"  
name : "Test Project"
```

## Project Detail Screen



## Dashboard Screen



# Challenges

- Reimplementing Registration
  - Some faulty logic with DB dependance was found
  - Incorrect data was pushed for Ethan's new profile screen
- Implementing Create Project screen
  - Connecting data from create project to DB
  - Making sure info in DB worked with Project details page
- Making sure group is on track and everyone is on same page

# Solutions

1. Removed and refactored large chunks and code and even an entire page to streamline logic and make registration more reliable
2. Working with Ethan to make sure his frontend pages had the DB info they needed and that correct information was pushed
3. Making sure group is meeting regularly and everyone is contributing



# Personal Accomplishments

- Learning Flutter, Dart, and Firebase
- Implemented features central to Database connectivity
- Read and work with others code to get database integration
- Taking on role of Team lead



# Group Summary

Nate Reinboldt

## Major Accomplishments

- Developed a large number of features for the small amount of time given
- Provide user login to authenticate and show personalized data
- Implemented a premium feature that allows user to create projects
- Worked together as a team without clear direction from a client

## Minor Accomplishments

- Learned Dart, Flutter, and Firebase
- Created and maintained file structure and encapsulation for readability and access.
- Gained experience using Git development environment

# Contribution Summary

## Anirudha Kyathsandra

- Project idea.
- Original UI design documents.
- Initial profile page implementation.
- Added features to project description, project creation.
- Premium account side of the project.

## Ethan Lindsey

- Designed Dashboard, Profile, and Settings.
- Integrated users and projects with Backend.
- Refactoring and Widget organization.
- Enhanced performance and security.

## Ishan Banerjee

- Project UI sketchup and design
- Screen routing and navigation system
- Login and signup registration
- Project Description page
- Chat UI and backend integration

## Nathan Reinboldt

- Created Sign in, Sign up, and Authentication Screens
- Created Project Create Screen
- Connected Frontend data to database



AntHill