



SkillCourt Summer 2022

CIS 4951 - Capstone II

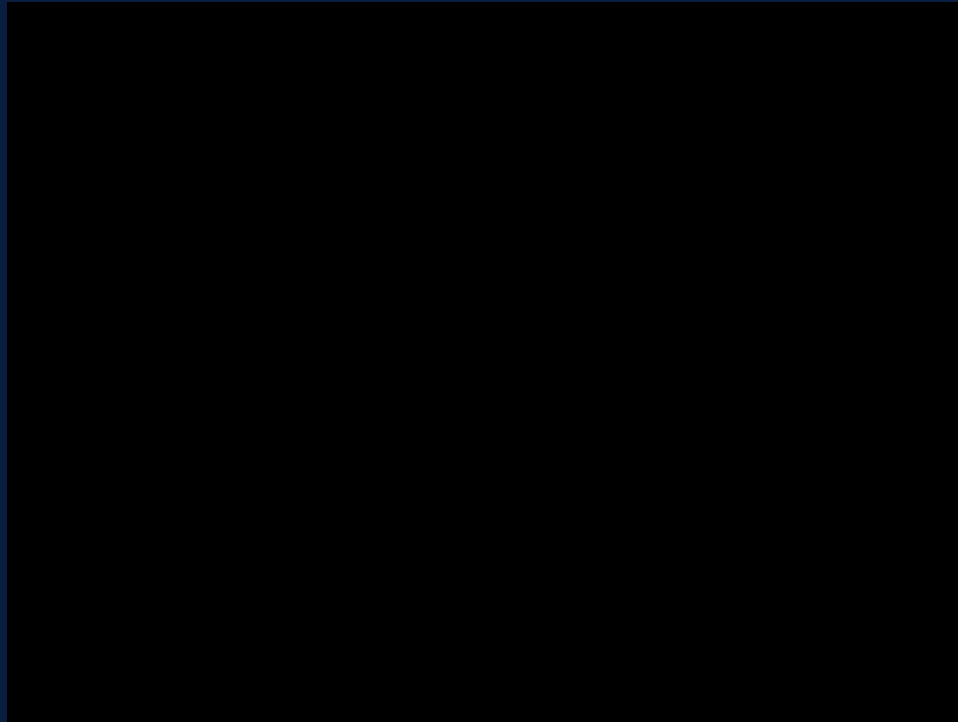
Knight Foundation School of Computing and Information
Sciences

Introduction

SkillCourt is a game where players are able to throw/kick/toss a ball at a sensor pad and then have the results of this toss recorded on their phone application. A typical SkillCourt setup involves multiple pads and lights up the pads in an order that is specified by the phone application.



Project Video





Project Goal (Mobile)

- Random pad game
- Connect the phone login system to the website
- Synchronize user data between website and phone app





Contributions (Mobile)

Mobile:

- In the previous semester, it was not possible to connect to multiple pads simultaneously
- Can now connect to multiple pads
- Created a random pad game
- Fixed rendering errors of widgets across the entire application





Project Goal (Website)

- The goal for the website team was to change the branding for project and update any information regarding old project
- The second goal for the website team was to work on server side of the website and connect mobile application to web server





Contributions (Website)

- The Website Team was able to change all branding regarding Agineek to Skillcourt in Website and code
- The Website Team was able to add mutations regarding server configuration But not enough testing to move with connecting mobile application to website.



Architecture (Mobile)

Backend:

- Written in nest JS

Frontend:

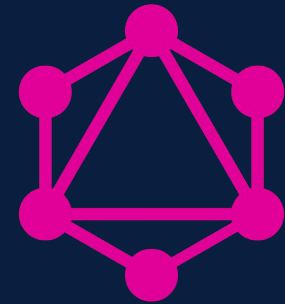
- Written entirely in Flutter
- Pad emulator written in Python



Architecture (Website)

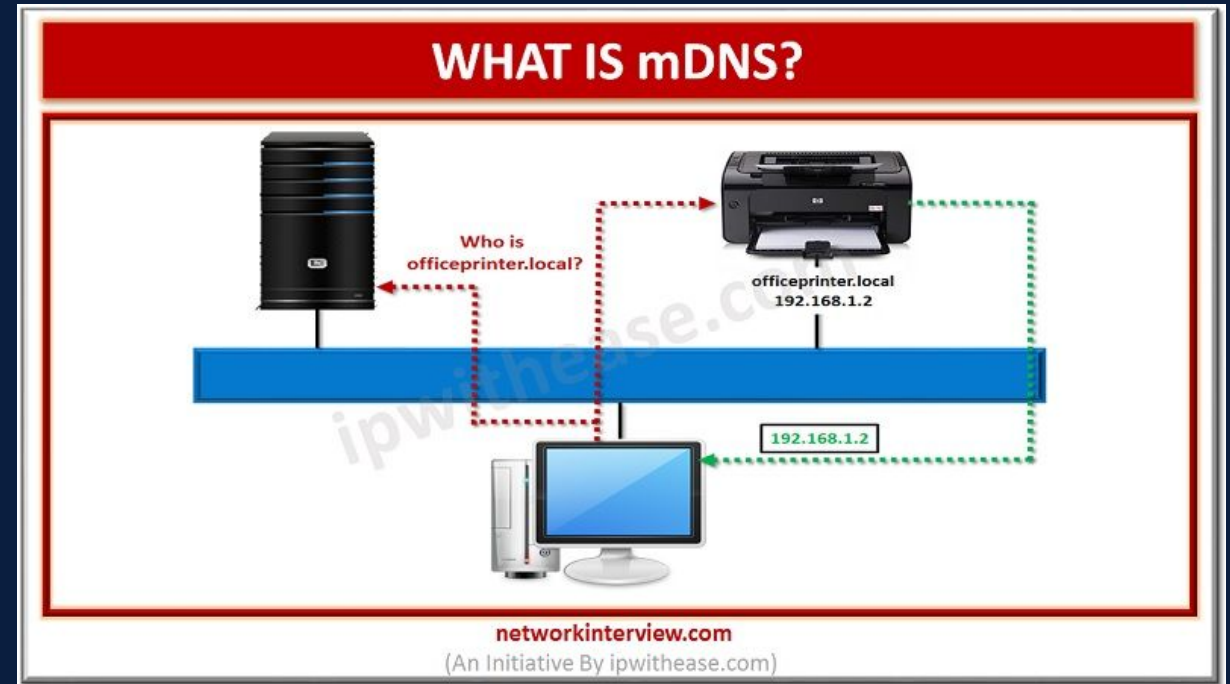
Backend:

- Node.js
- Typescript
- GraphQL
- Docker



MDNS and DNS-SD

- Using the flutter_nsd package we can scan for registered services on the local network.
- The pads register themselves on the network.
- Can connect to multiple pads simultaneously using flutter Isolates.
- Send codes to pads to control their functionality.

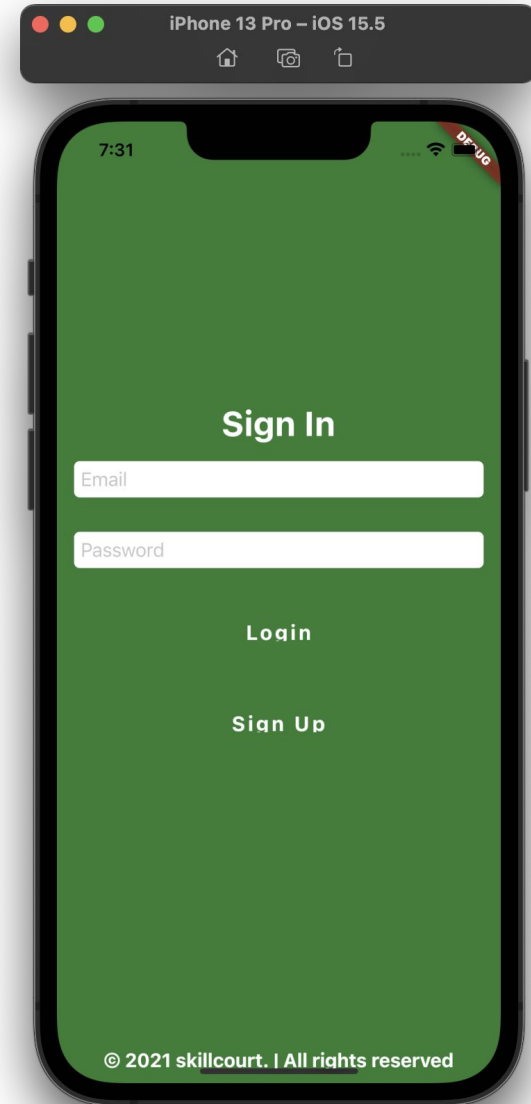
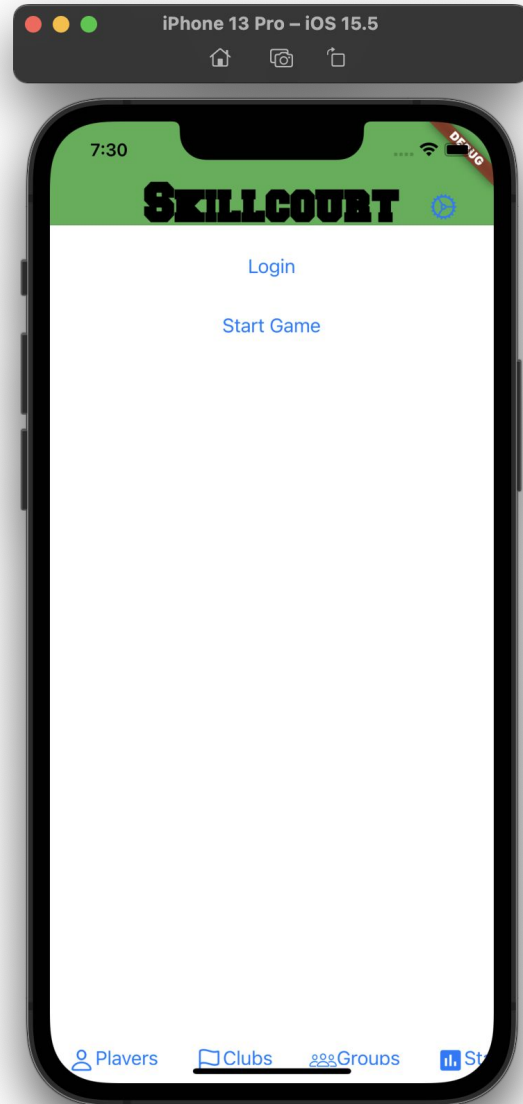


SkillCourt Phone App v2

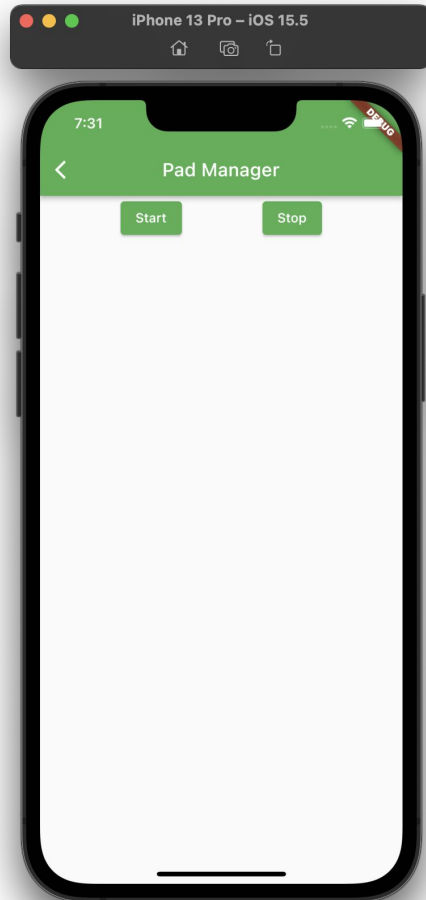
- Number of Commits: 50
- Work:
 - Dart Application



SkillCourt Phone App



SkillCourt Phone App – Pad Manager



```
class PadManagerPage extends StatefulWidget {
  static const routeName = 'pad-manager';

  const PadManagerPage({Key? key}) : super(key: key);

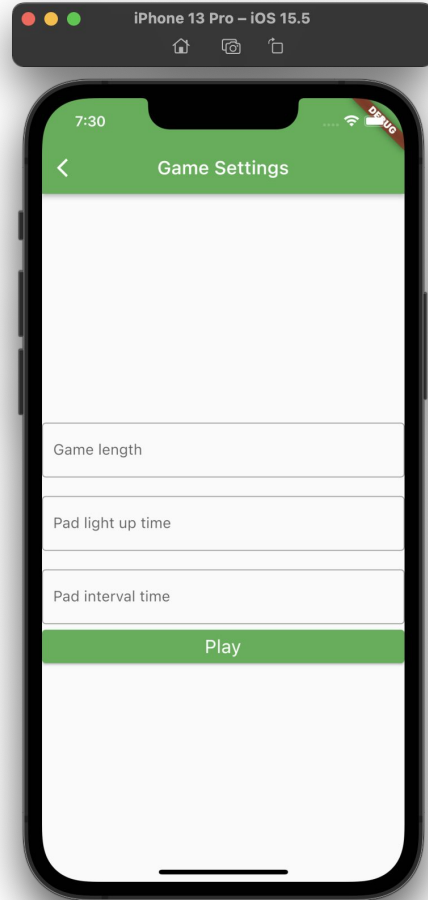
  @override
  State createState() => _PadManagerPageState();
}

class _PadManagerPageState extends State<PadManagerPage> {
  final flutterNsd = FlutterNsd();
  late String padResult;
  late Event hasResult;
  List<NsdServiceInfo> services = <NsdServiceInfo>[];
  Map<NsdServiceInfo, Pad> socketsMap = <NsdServiceInfo, Pad>{};

  bool initialStart = true;
  bool _scanning = false;

  _PadManagerPageState();
}
```

SkillCourt Phone App



```
Future<void> startGame(BuildContext context) async {
  socketMap = padManagerPage.getMap();
  services = padManagerPage.getServices();
  //set counter of hits to 0
  padManagerPage.setHits();

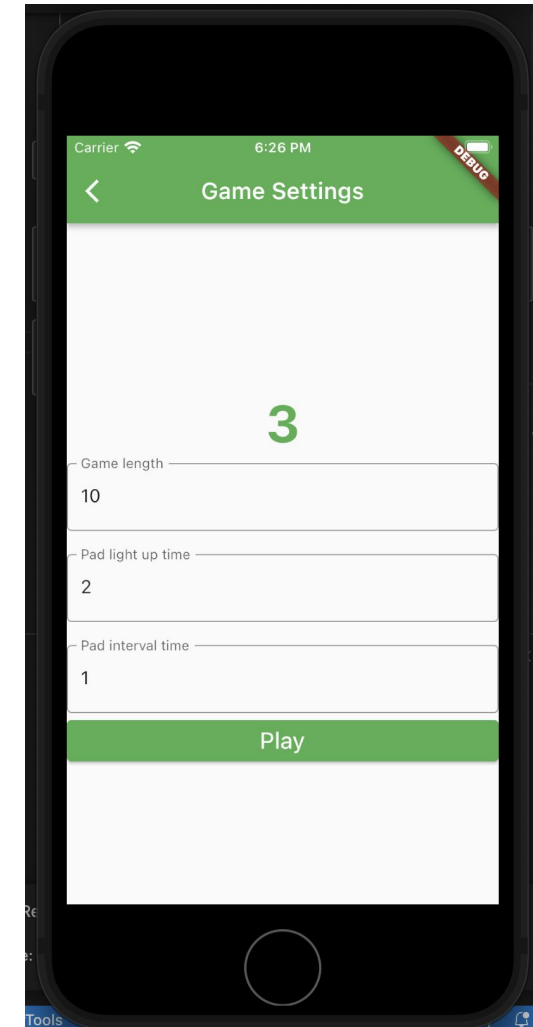
  final int range = services.length;

  for (int i = 0; i < range; i++) {
    final Pad pad = socketsMap[services[i]]!;
    listOfPads.add(pad);
  }
  final Random rand = Random();
  final int maxNumHits = (_gameLen / (_padInterval + _padLightUp)).ceil();
  int seconds = 0;
  // ignore: unused_local_variable
  Timer _timer;
```

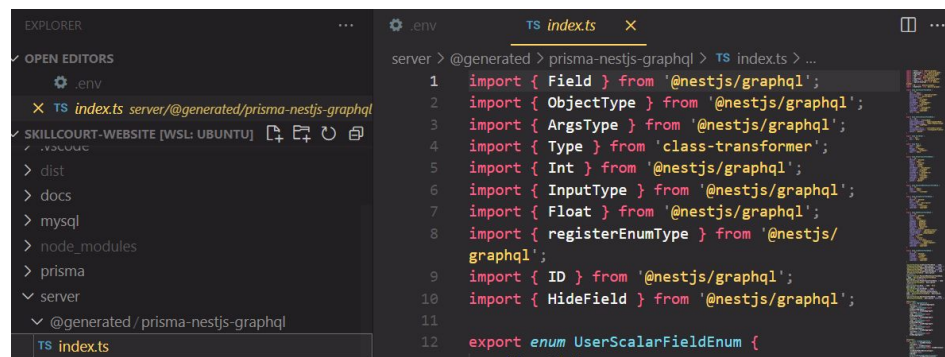
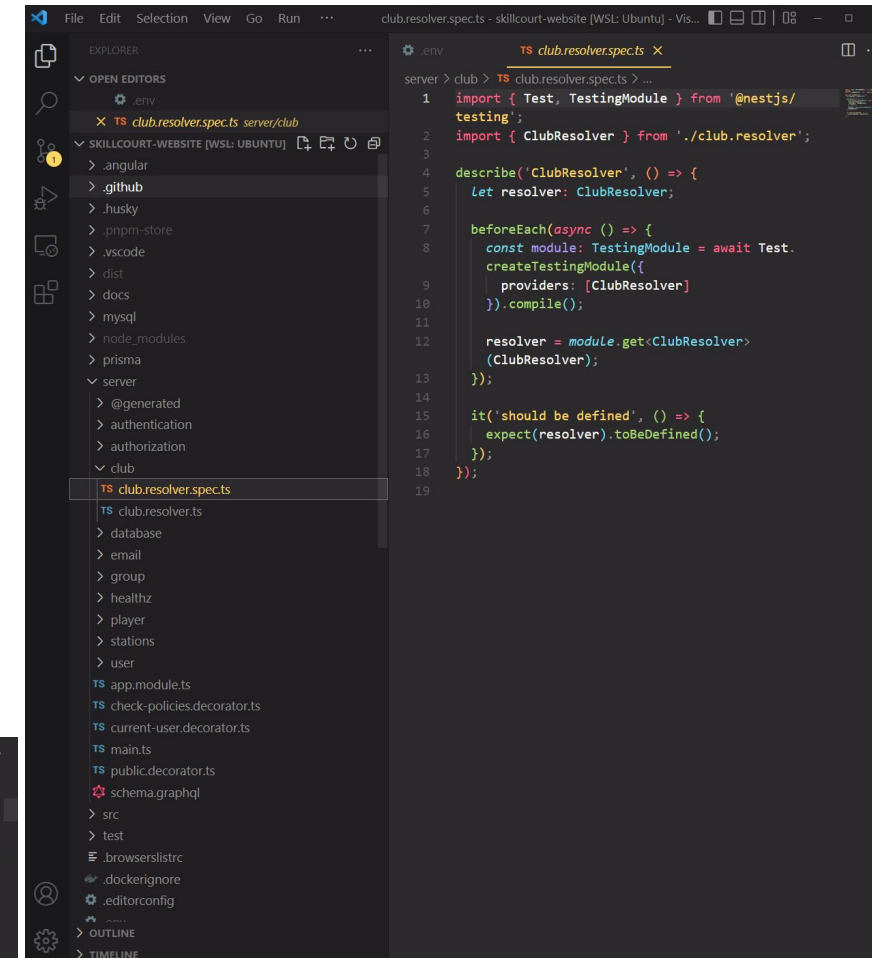
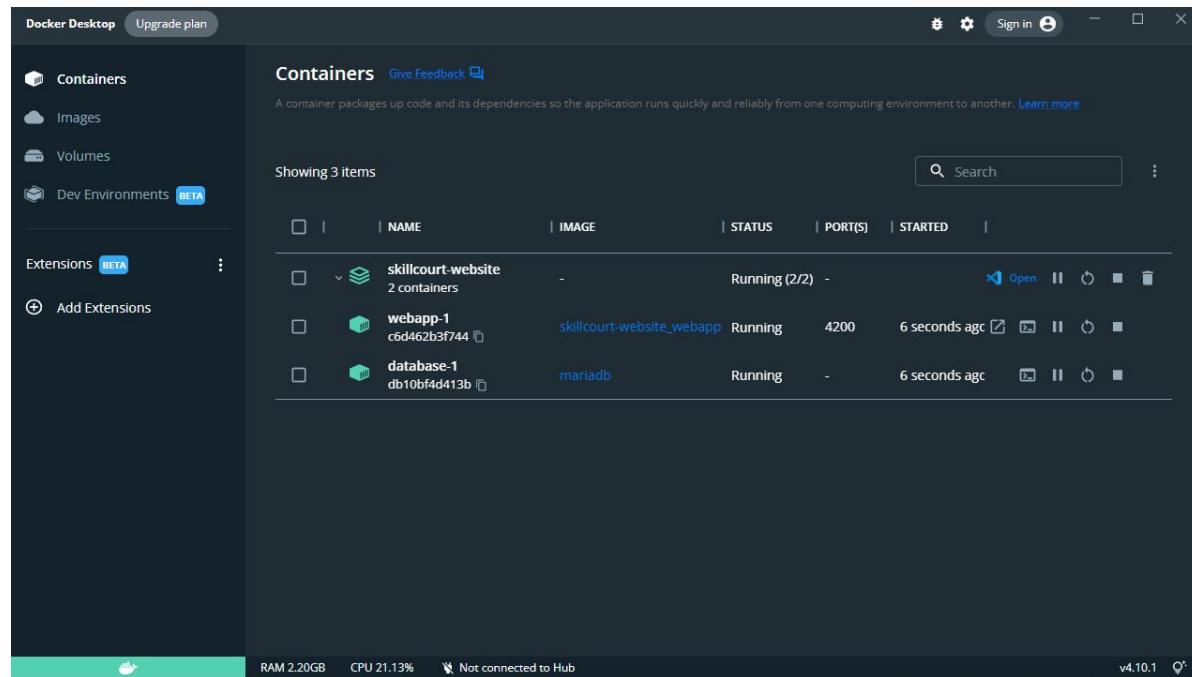
SkillCourt Phone App v2

```
_timer = Timer.periodic(const Duration(seconds: 1), (timer) {  
  timerOn = true;  
  seconds = seconds + 1;  
  // _gameLen = _gameLen - 1;  
  setState(() => _gameLen--);  
});
```

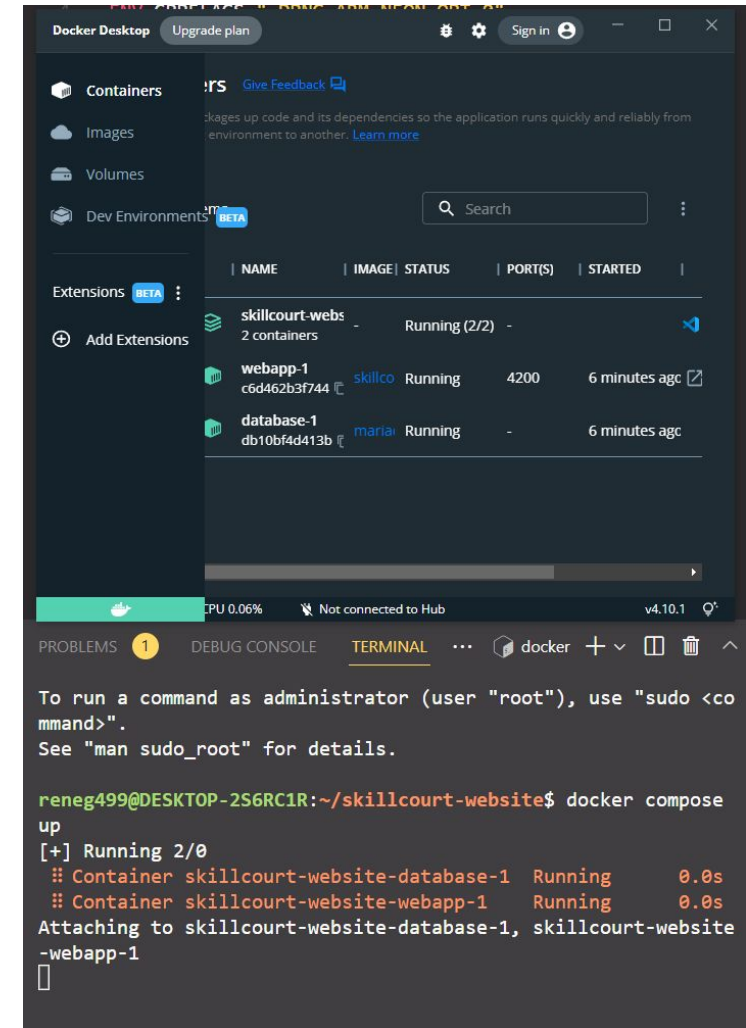
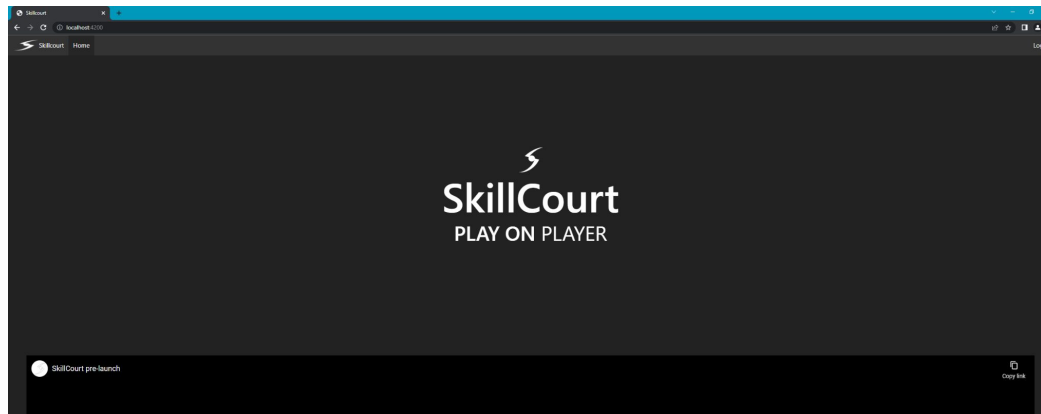
```
if (timerOn)  
  Text('$gameLen',  
    textAlign: TextAlign.center,  
    style: const TextStyle(  
      fontWeight: FontWeight.bold,  
      height: 2,  
      color: Colors.green,  
      fontSize: 45,  
    )) // TextStyle // Text  
else  
  const Text(' '),  
  TextField(  
    onSubmitted: (value) {  
      _gameLen = int.parse(value);  
    },  
    decoration: const InputDecoration(  
      border: OutlineInputBorder(),  
      labelText: 'Game length',  
    )), // InputDecoration // TextField
```



SkillCourt Website



SkillCourt Website



SkillCourt Website



About
Works
Training
Contact



ABOUT SKILLCOURT

Skillcourt makes wireless stand-alone panels that run on batteries and display a matrix of LED lights. A typical set up is to have 3 or more panels during play. Each panels acts as a target for Soccer players to strike with a ball, The ball then rebounds back to the player while the players have to figure out the most efficient way to maneuver the ball and get to the next illuminated

FIU

Engineering
& Computing

FLORIDA INTERNATIONAL UNIVERSITY

Project Requirements

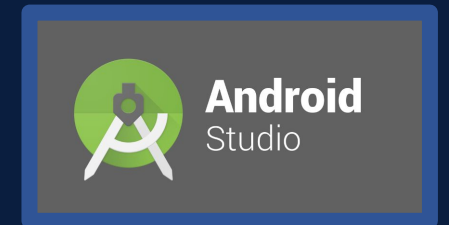
- In order to work on this project, the proper environment is needed. The following is necessary:

Website

- Node version 14.0.0
- Pnpm version 6.21.0
- Docker
- Install WSL2 and use a proper distro
- Mac Vagrant
- Mac Virtual Box

Mobile

- Android Studio (for windows) or XCode (for MacOS)
- git
- git-flow or another git-flow plugin for Git.
- A proper emulator for Android or iOS
- Flutter SDK along with Dart
- Java JRE version 1.8 or greater
- Python 3 (for the pad emulator)



Shortcomings

- We were unable to:
 - Connect phone login system to website.
 - Connect synchronization service to website.

```
#0 6.648 npm notice
#0 6.648 npm notice New minor version of npm available! 8.11.0 -> 8.14.0
#0 6.649 npm notice Changelog: <https://github.com/npm/cli/releases/tag/v8.14.0>
#0 6.649 npm notice Run `npm install -g npm@8.14.0` to update!
#0 6.649 npm notice
#0 7.172 ERROR Unable to find the global bin directory
#0 7.172 Run "pnpm setup" to create it automatically, or set the global-bin-dir setting, or the PNPM_HOME env variable. The global bin directory should be in the PATH.
-----
failed to solve: executor failed running [/bin/sh -c apk add zlib zlib-dev optipng pkgconfig autoconf automake libtool nasm build-base python3 openssl && npm i -g pnpm && pnpm i -g @angular/cli prisma && chown -R node:node /app]: exit code: 1
```





SkillCourt Website (Shortcomings)

- Server/docker problems for a big chunk of the sprints the website team was having problems with docker and the containers they could not connect to the server preventing any progress in the website side.
- New Coding Languages: Not all members of the group were familiar with the language they were coding in. Some members needed to learn Swift or Typescript depending on what platform they were developing for. This consumed a lot of development time

