

INSTRUCTIONS:

Goal of the Project:

In Class 41, you revised all the concepts learned in the Speed Racer game so far. You also learned how to create obstacles to make the game more difficult. In this project, you have to practice what you learned in the class to make some obstacles for the Fruit Catcher game.

**** This is a continuation of Project 39, Project 40 so make sure to complete that before doing this project. ****

Story:

Honey visited her grandparents, where there was a farm. Farmers were cutting fruits to harvest them. Her grandfather gave her the responsibility to collect these harvested fruits in a basket. Honey went with her cousin to the farm to collect the fruits. She really had fun collecting them and on her way back, she thought of creating an app to play with her cousin. To make the game more interesting she has added the game obstacles.

Let's see who collects more, Honey or her cousin. Here, you will make the game more challenging in this project by adding obstacles to the game.

Watch a video of this in action [here](#).

Project Template Output



Project Expected Output



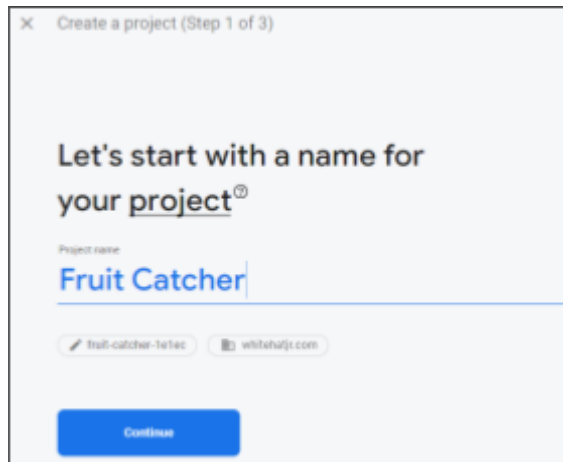
***This is just for your reference. We expect you to apply your own creativity to the project.**

Getting Started:

1. Download the project from this [link](#).
2. **Unzip** the folder.
3. **Rename** the unzipped folder as **Project 41**.
4. **Import** this folder into **VS Code**.
5. Start making changes

Specific Tasks to complete the Project:

1. Setup Firebase for the Project.
 - Go to your [Firebase console](#) and click on **Create a Project**.
 - Enter the name of the Project as **Fruit Catcher Game**.
 - Accept terms and click on **Continue**.



✕ Create a project (Step 1 of 3)

Let's start with a name for your project®

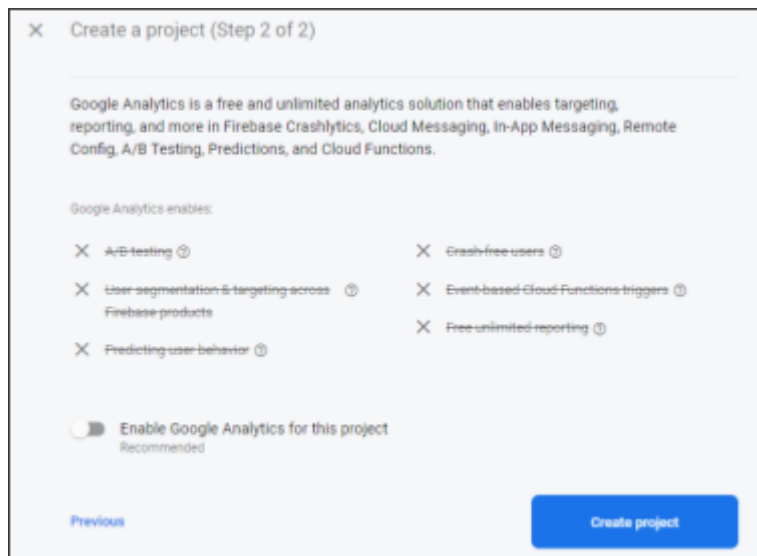
Project name

Fruit Catcher

fruit-catcher-1e1ec whitehatjr.com

Continue

- Disable the Google Analytics option
- Click on **Create Project**



✕ Create a project (Step 2 of 2)

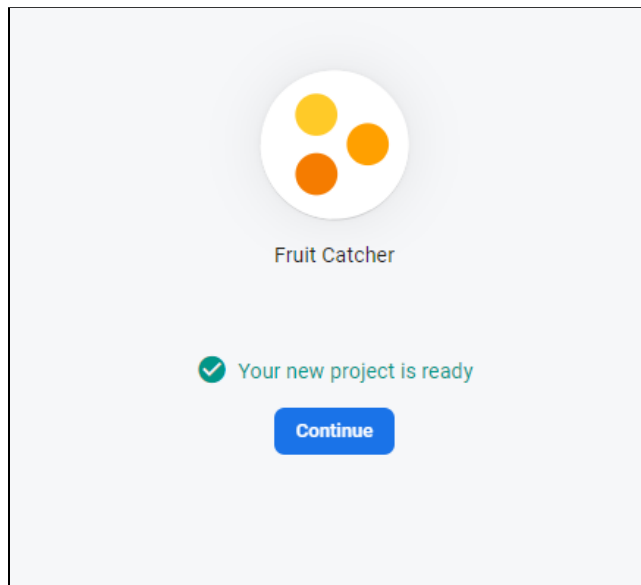
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

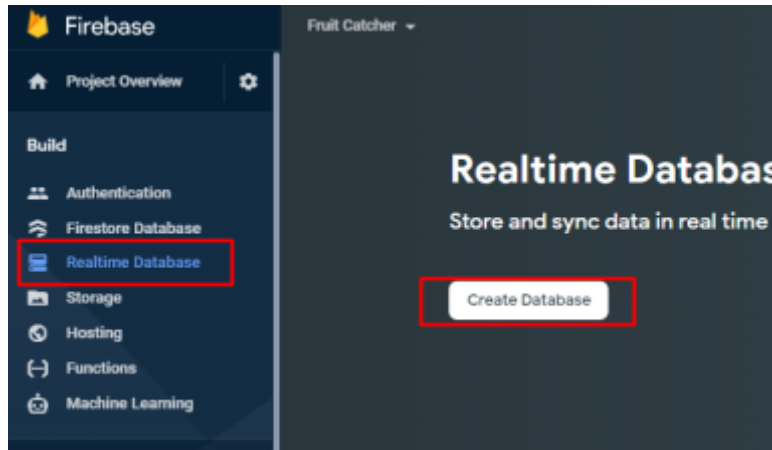
✕ A/B testing ⓘ	✕ Crash-free users ⓘ
✕ User segmentation & targeting across Firebase products ⓘ	✕ Event-based Cloud Functions triggers ⓘ
✕ Predicting user behavior ⓘ	✕ Free unlimited reporting ⓘ

☐ Enable Google Analytics for this project
Recommended

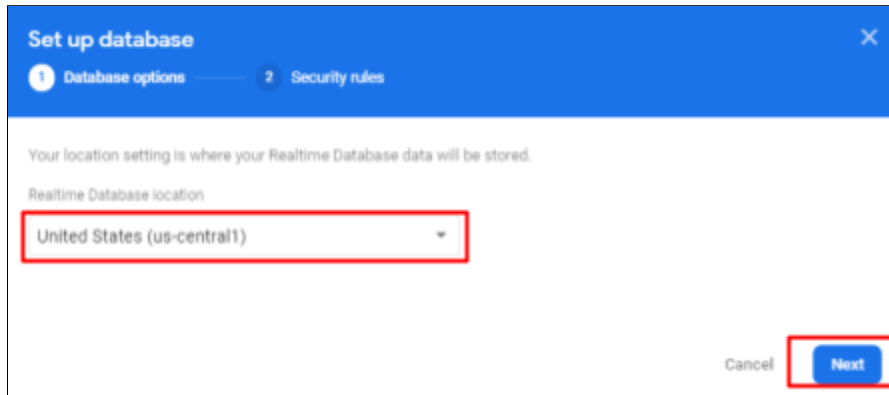
Previous Create project



- On the left-hand side panel, click on **BUILD**
- Click on **Real-time Database** and then click on **Create Database**



- Select **United States (us-central1)** in the **Setup Database** window.



Set up database

1 Database options 2 Security rules

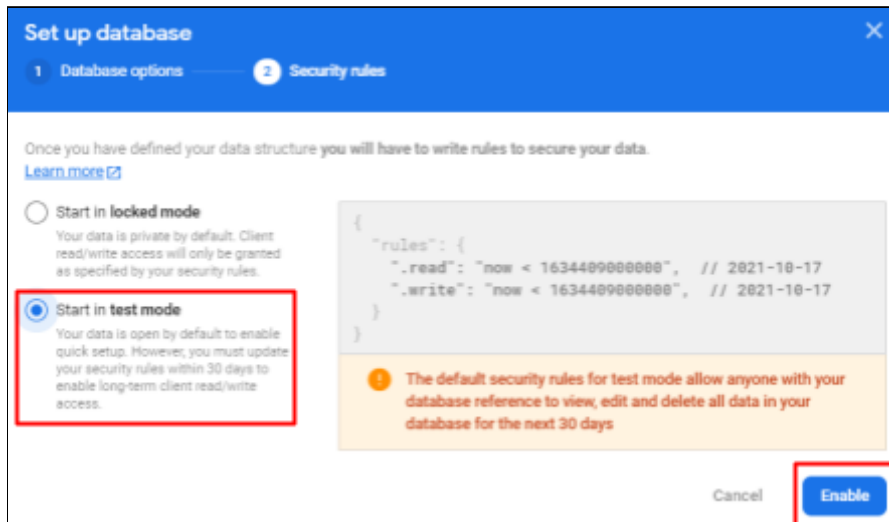
Your location setting is where your Realtime Database data will be stored.

Realtime Database location

United States (us-central1)

Cancel **Next**

- To create a database in test mode, click on **start in test mode** and click on **Enable**.



Set up database

1 Database options 2 Security rules

Once you have defined your data structure you will have to write rules to secure your data.
[Learn more](#)

☐ Start in locked mode
 Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☒ **Start in test mode**
 Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```

{
  "rules": {
    ".read": "now < 1634489800000", // 2021-10-17
    ".write": "now < 1634489800000", // 2021-10-17
  }
}

```

1 The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel **Enable**

- Create a node in the **Database** as a **gameState**, **playerAtEnd**, and **playerCount**. Set **0** as the value for both the nodes.

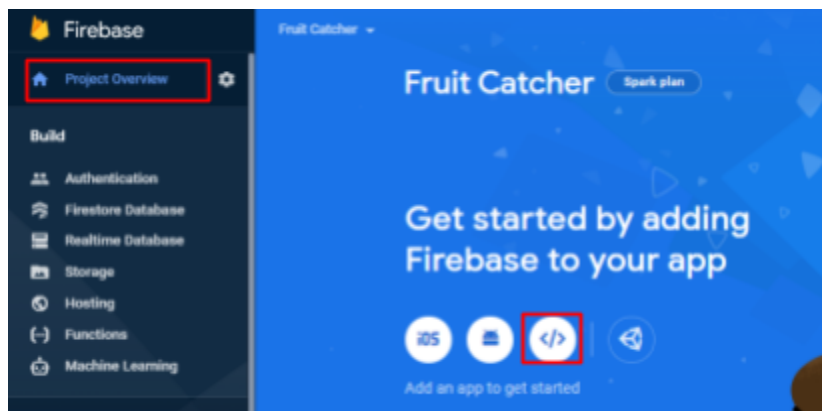
fruit-collector-1cb1a-default-rtdb

gameState: 0

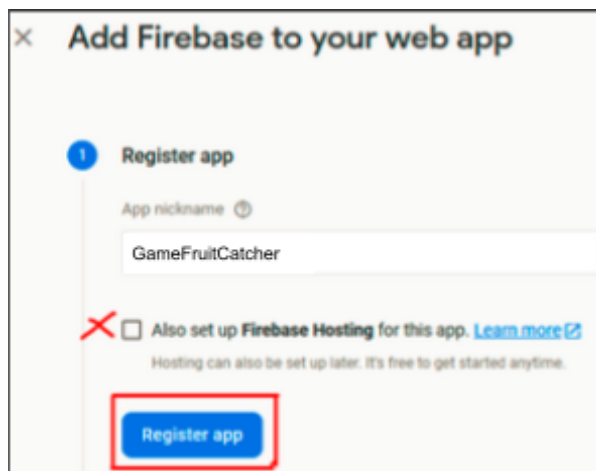
playerAtEnd: 0

playerCount: 0

- On top of your left under **Project Overview**, and select the **Web** option.





- Register the app and **don't check** the Firebase hosting option.



- Add Firebase **SDK**.
 - Copy the content by clicking on the icon to the bottom right and paste it into the **index.html** file along with an src library for the Firebase database in VS.

2 Add Firebase SDK

☒ Use npm  ☐ Use a <script> tag 

If you're already using [npm](#) and a module bundler such as [webpack](#) or [Rollup](#), you can run the following command to install the latest SDK:

```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyO_rdrav0oMj0thaktR9qw7igsAz6bKSI8",
  authDomain: "fruit-catcher-1elec.firebaseio.com",
  databaseURL: "https://fruit-catcher-1elec-default-rtdb.firebaseio.com",
  projectId: "fruit-catcher-1elec",
  storageBucket: "fruit-catcher-1elec.appspot.com",
  messagingSenderId: "345854395863",
  appId: "1:345854395863:web:a56776bec4b456cc8d1f92"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```



Note: This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

Continue to console

Specific Tasks to complete the Project: Specific Tasks to complete the Project:

The following table highlights instructions to be added to the project template. For each task, the code blocks are commented on, you have to **uncomment the correct block of code**.

Things to do:	Steps / Final output
<div> <div>Step 1</div>  <p>In game.js, in function addObstacles() uncomment the correct line of code to create an obstacle at random x position.</p> </div>	<pre>addObstacles() { var x, y; //x= 200; //x = random(0, width-100); //x = random(0); //x = random(); }</pre>
<div> <div>Step 2</div>  <p>In game.js, in function play() uncomment the correct line of code to assign value to gameState to end the game when the obstacleGroup touches the players.</p> </div>	<pre>if(obstacleGroup.isTouching(players)){ // gameState = 0; // gameState = 1; // gameState = 2; // gameState = 3; }</pre>



Submitting the Project:

1. **Upload** your completed project to your **GitHub** account. **Here is a video on how to do this:** <https://vimeo.com/561338335/aa2b0db66e>
2. Enable **GitHub** pages for the repository. After you have done this step, wait for a few minutes for the website for your project to be live. **See the video given below:** <https://vimeo.com/561338446/a7e3084fb4>
3. Copy the link to the GitHub pages link in the Student Dashboard.

REMEMBER... Try your best, that's more important than being correct.

After submitting your project your teacher will send you feedback on your work.

————— xxx ————— xxx ————— xxx ————— xxx ————— xxx —————