Front-End Development

React & Firebase

#2 Firebase Database & Storage

Purwadhika
Startup and Coding School

How to Connect React Project to Firebase Database

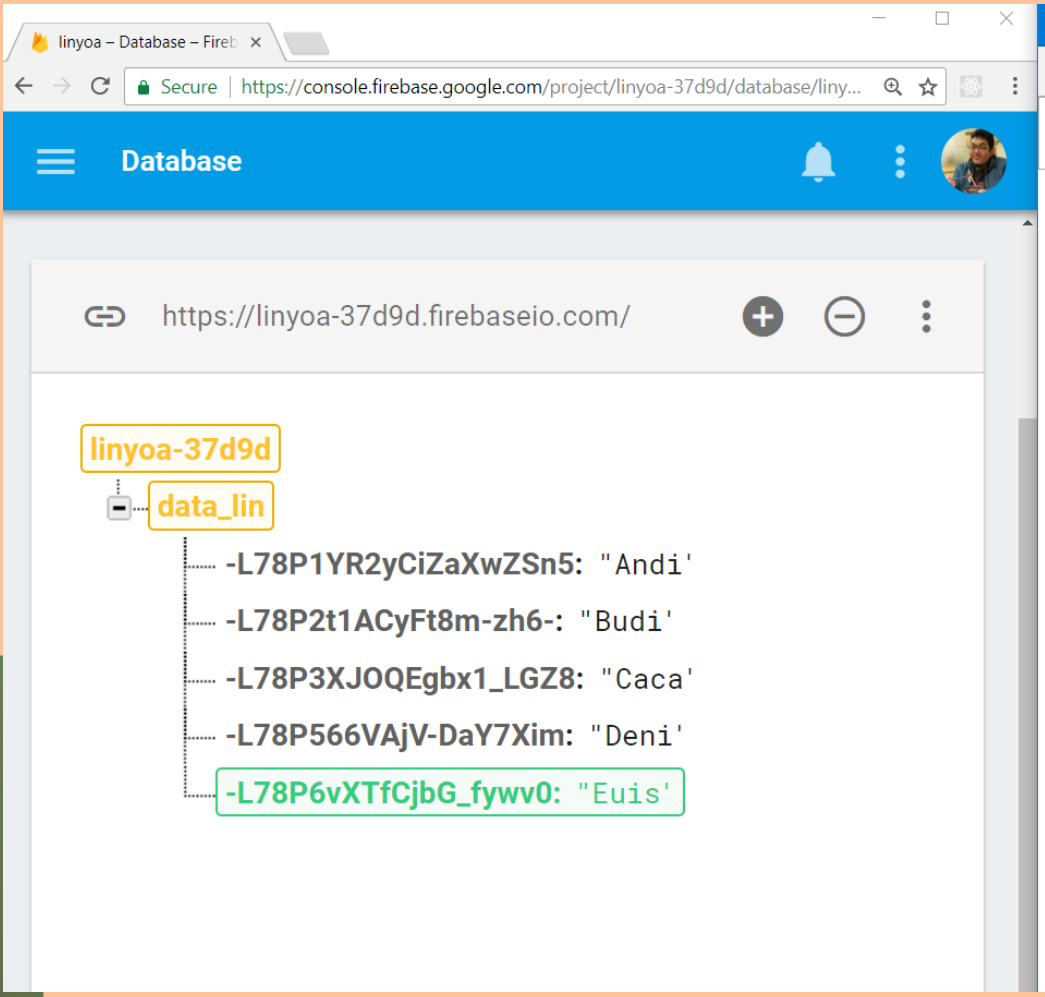
Purwadhika
Startup and Coding School

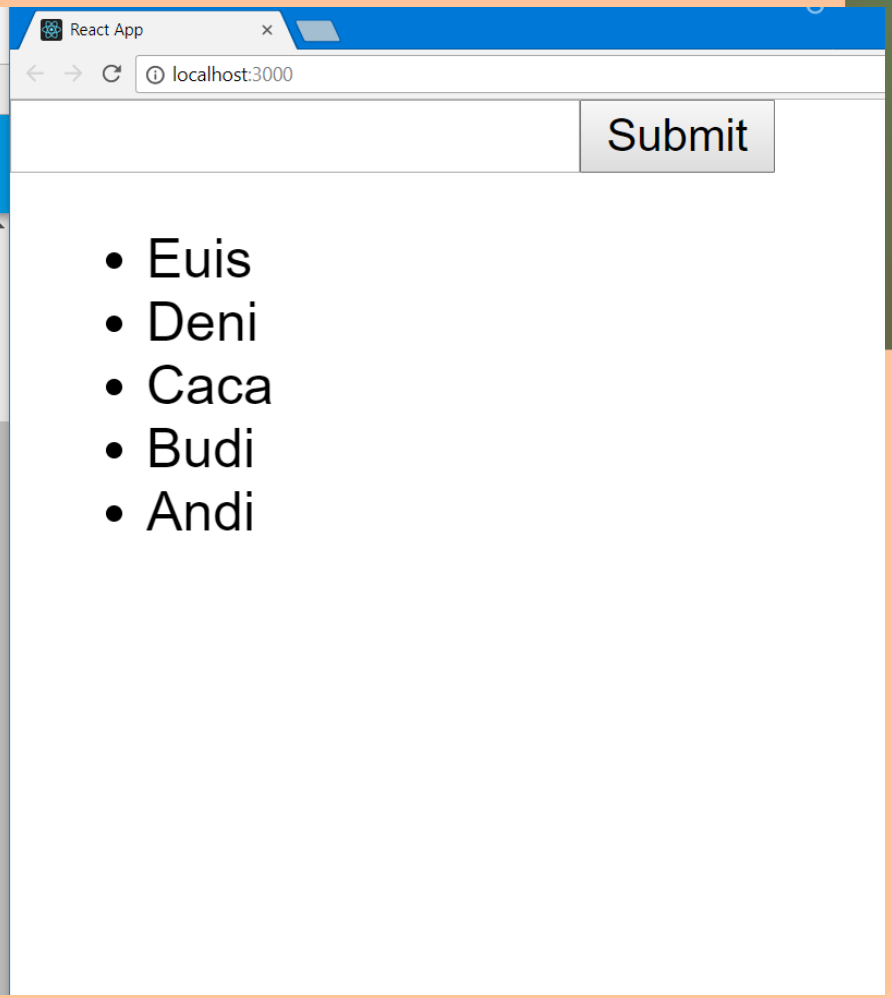
Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every client.

When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Purwadhika
Startup and Coding School

React & Firebase Database





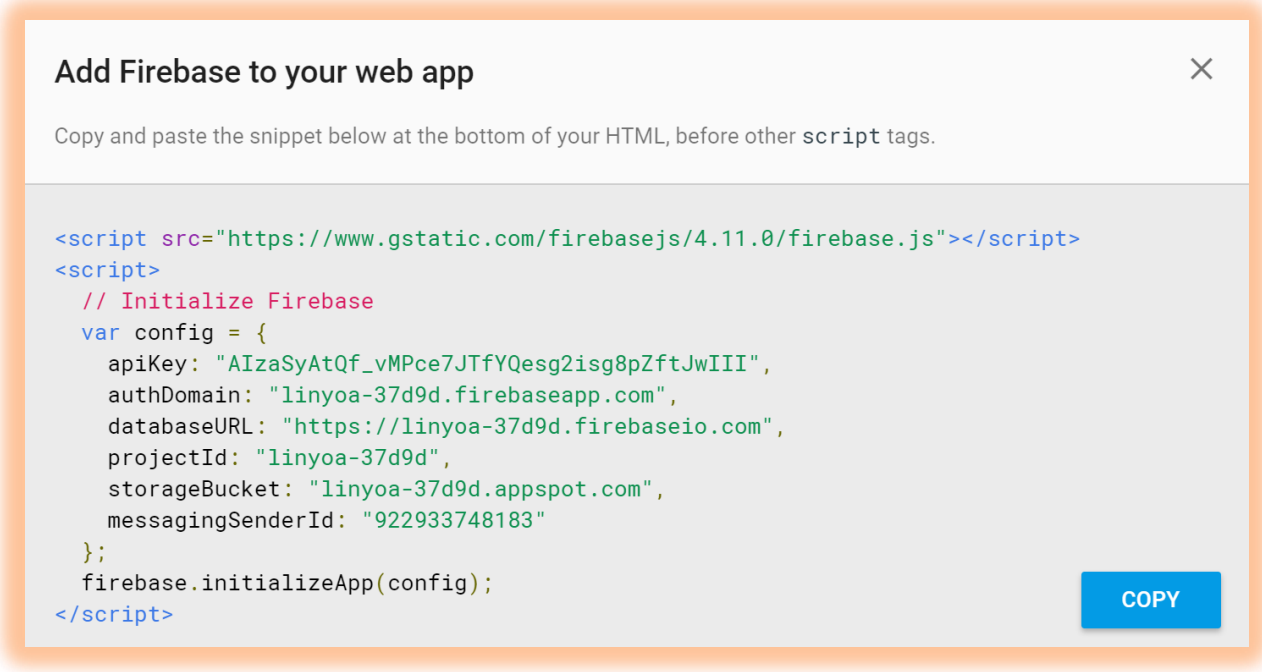
Purwadhika
Startup and Coding School

#1 Setup Firebase Database

- On your project dir, install Firebase Package!

```
$ npm install firebase --save
```

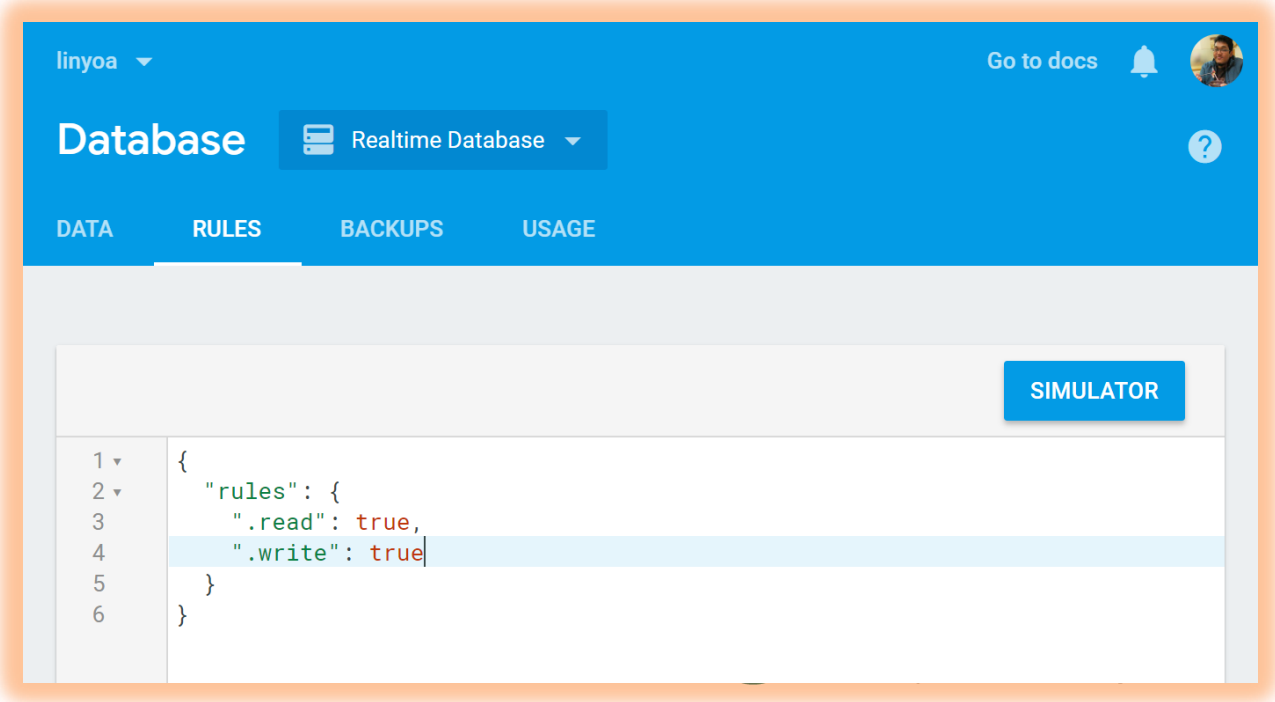
- On Firebase Database project, click ***Project Overview → Add Firebase to Your Web App*** then copy its snippet code.



#1 Setup Firebase Database

- On Firebase Database Rules (also on JSON file *database.rules.json* **if any**), set all its value to *true*:

```
{
      "rules": {
            ".read": true,
            ".write": true
      }
}
```



#2 Create/Prepare Your React App

- Install create-react-app utility globally on your PC (if you have not done yet before)

```
$ npm install -g create-react-app
```

- Create a new react app or prepare yours

```
$ create-react-app linyoa
$ cd linyoa
```

- Make sure that it runs successfully

```
$ npm start
```

Purwadhika
Startup and Coding School

#2 Create/Prepare Your React App

- Install Firebase packages needed:

```
$ npm install -g firebase-tools
$ npm install firebase
```

Purwadhika
Startup and Coding School

src/fire.js

#3 Insert Snippet

Create src/fire.js

```
import firebase from 'firebase';

var config = {
        apiKey: "a1b2c3d4e5f6g7h8i9j0",
        authDomain: "linyoa-37d9d.firebaseapp.com",
        databaseURL: "https://linyoa-37d9d.firebaseio.com",
        projectId: "linyoa-37d9d",
        storageBucket: "linyoa-37d9d.appspot.com",
        messagingSenderId: "1234567890"
};

var fire = firebase.initializeApp(config);

export default fire;
```

Purwadhika
Startup and Coding School

src/App.js

#4 Set Reference to Firebase Database

```
import React, { Component } from 'react';
import fire from './fire';

class App extends Component {
 constructor(props) {
  super(props);
  this.state = { messages: [] };
}

componentWillMount(){
 let messagesRef =
fire.database().ref('data_lin').orderByKey().limitToLast(100);
 messagesRef.on('child_added', snapshot => {
 let message = { text: snapshot.val(), id: snapshot.key };
 this.setState({ messages: [message].concat(this.state.messages)
 });
})
}
```

Purwadhika
Startup and Coding School

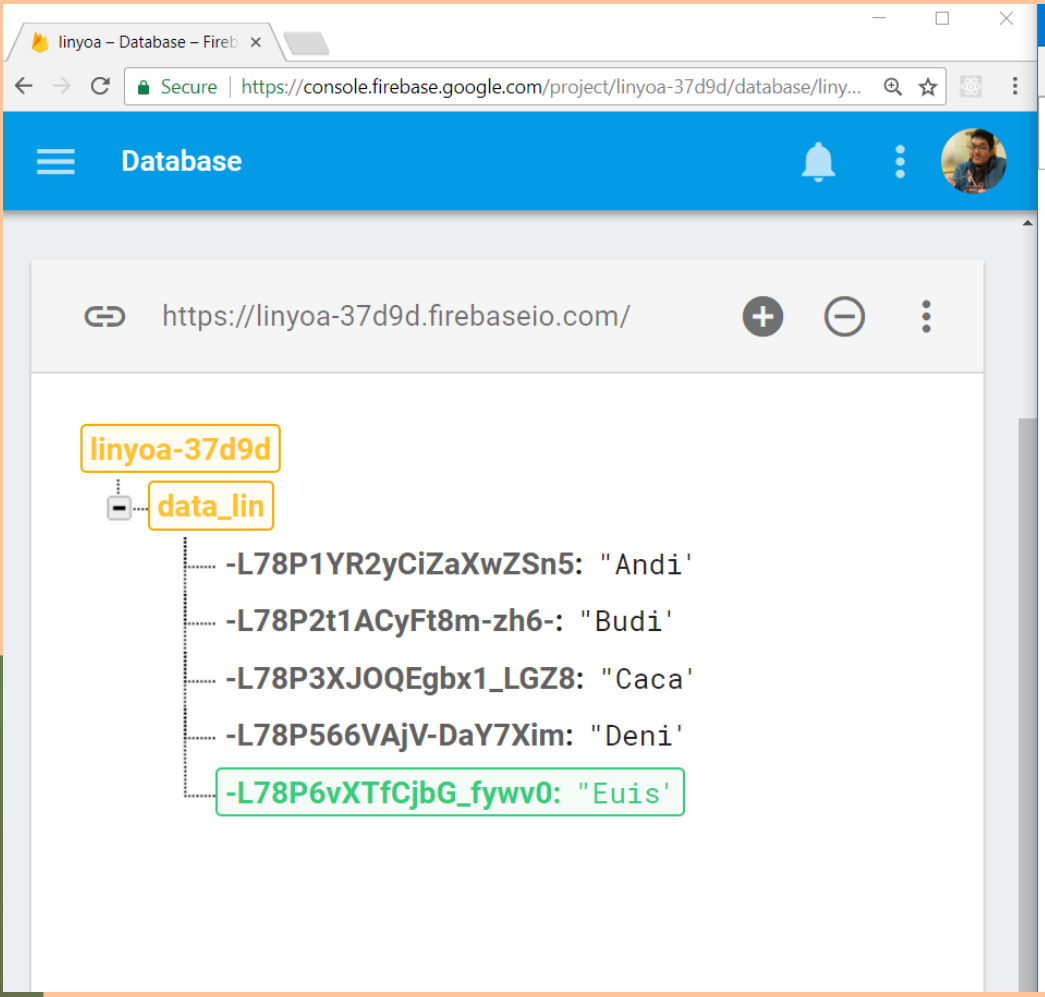
src/App.js

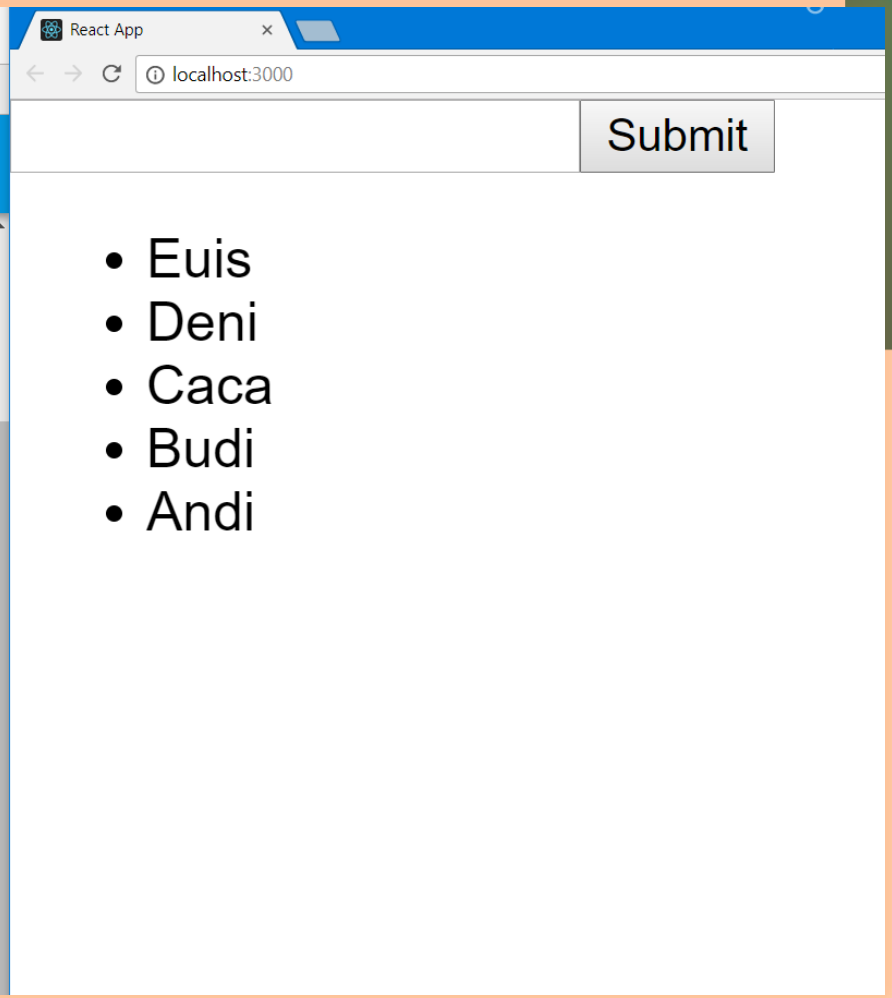
#4 Set Reference to Firebase Database

```
addMessage(e){
 e.preventDefault();
 fire.database().ref('data_lin').push( this.inputEl.value );
 this.inputEl.value = '';
}

render() {
 return (
  <form onSubmit={this.addMessage.bind(this)}>
   <input type="text" ref={ el => this.inputEl = el }/>
   <input type="submit"/>
   <ul>
    {
     this.state.messages.map( message => <li
     key={message.id}>{message.text}</li> )
    }
   </ul>
  </form>
);}}

export default App;
```

Purwadhika
Startup and Coding School

React & Firebase Database



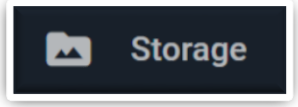


Purwadhika
Startup and Coding School

How to Connect React Project to Firebase Storage

Purwadhika
Startup and Coding School

Firebase

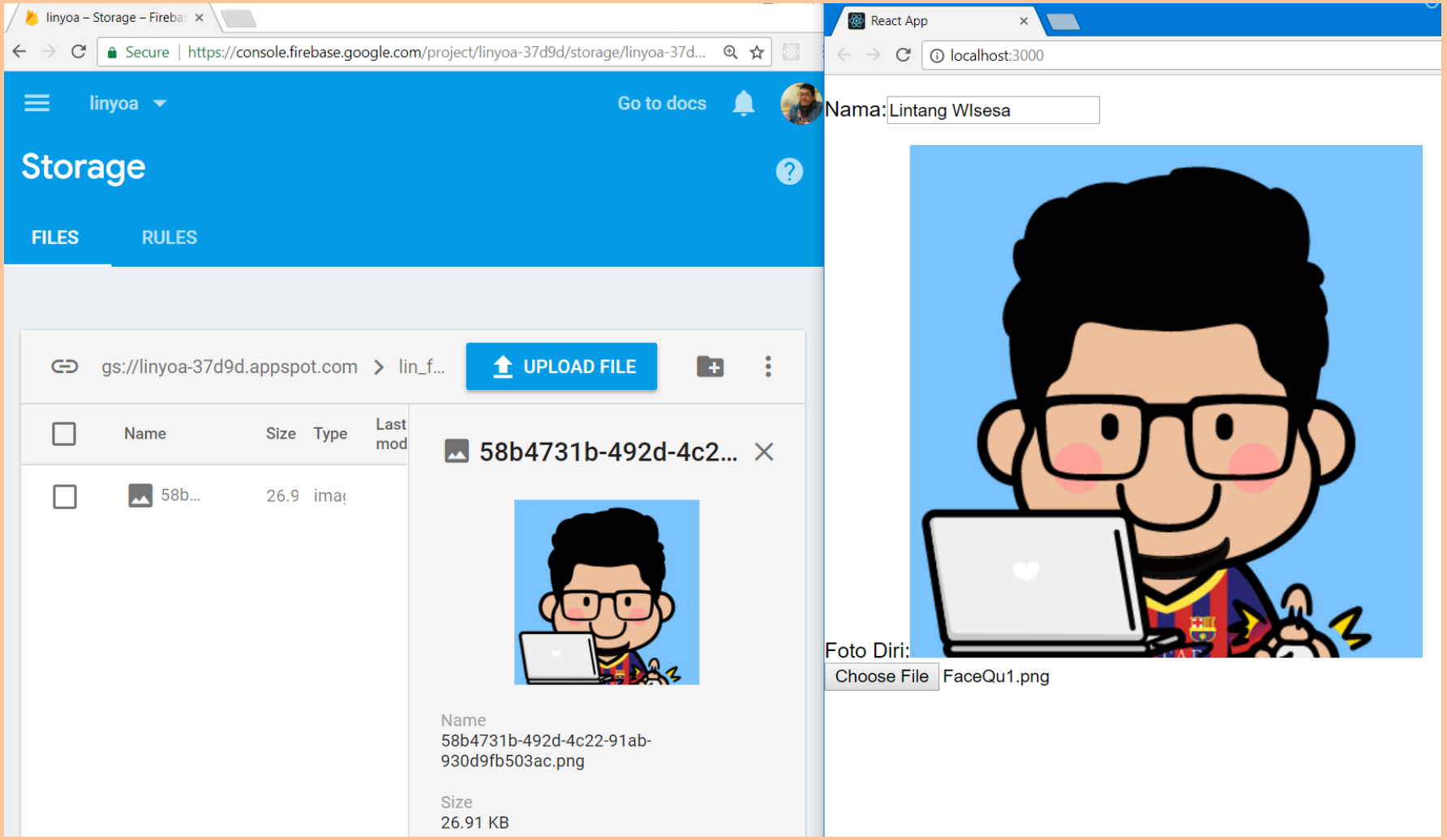


Firebase Storage is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.

You can use our SDKs to store images, audio, video, or other user-generated content. On the server, you can use **Google Cloud Storage**, to access the same files.

Purwadhika
Startup and Coding School

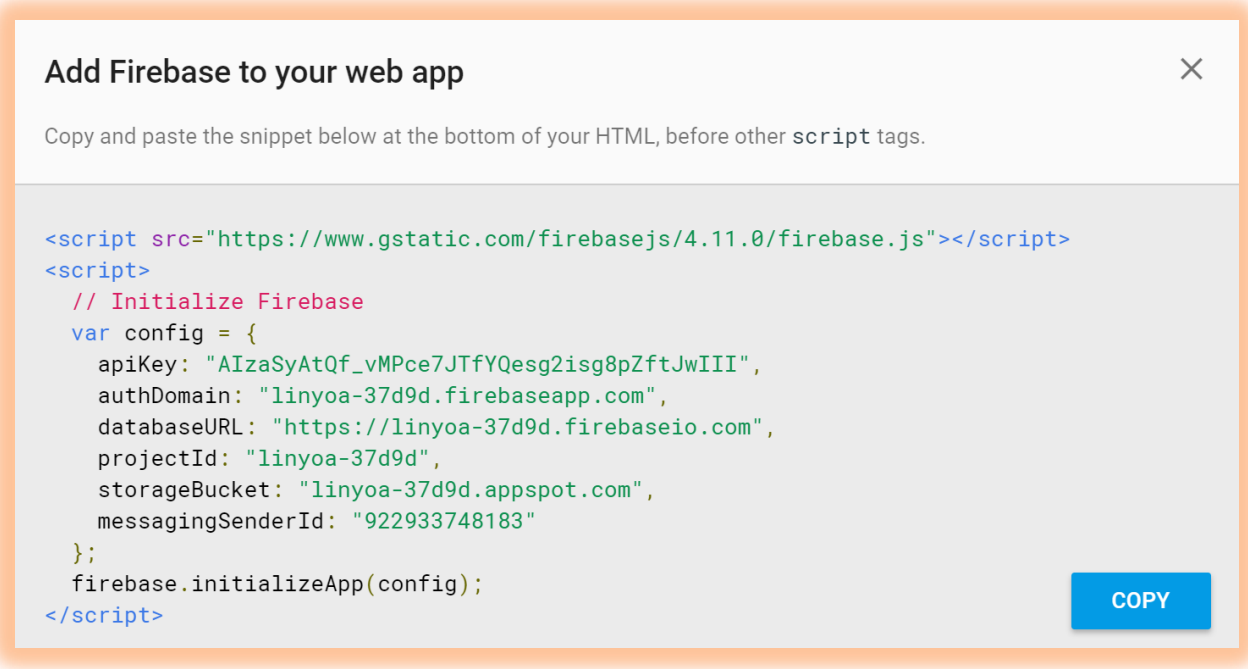
React & Firebase Storage



Purwadhika
Startup and Coding School

#1 Setup Firebase Storage

- Create a new Firebase project, or simply you can use your old project too.
- On Firebase project, click ***Project Overview*** → ***Add Firebase to Your Web App*** then copy its snippet code.



#1 Setup Firebase Storage

- On Firebase Storage Rules, turn off auth request by set its code as following, then publish it:

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth == null;
    }
  }
}
```

#2 Create/Prepare Your React App

- Install create-react-app utility globally on your PC (if you have not done yet before)

```
$ npm install -g create-react-app
```

- Create a new react app or prepare yours

```
$ create-react-app linyoa
$ cd linyoa
```

- Make sure that it runs successfully

```
$ npm start
```

Purwadhika
Startup and Coding School

#2 Create/Prepare Your React App

- Install Firebase packages needed:

```
$ npm install -g firebase-tools
$ npm install firebase
$ npm install react-firebase-file-uploader
```

Purwadhika
Startup and Coding School

src/fire.js

#3 Insert Snippet

Create src/fire.js

```
import firebase from 'firebase';

var config = {
        apiKey: "a1b2c3d4e5f6g7h8i9j0",
        authDomain: "linyoa-37d9d.firebaseapp.com",
        databaseURL: "https://linyoa-37d9d.firebaseio.com",
        projectId: "linyoa-37d9d",
        storageBucket: "linyoa-37d9d.appspot.com",
        messagingSenderId: "1234567890"
};

var fire = firebase.initializeApp(config);

export default fire;
```

Purwadhika
Startup and Coding School

src/App.js

#4 Set Reference to Firebase Storage

```
import React, { Component } from 'react';
import fire from './fire';
import firebase from 'firebase';
import FileUploader from 'react-firebase-file-uploader';

class ProfilePage extends Component {
state = {
username: '',
avatar: '',
isUploading: false,
progress: 0,
avatarURL: ''
};
```

Purwadhika
Startup and Coding School

src/App.js

#4 Set Reference to Firebase Storage

```
handleChangeUsername = (event) => this.setState({username:
event.target.value});

handleUploadStart = () => this.setState({isUploading: true,
progress: 0});

handleProgress = (progress) => this.setState({progress});

handleUploadError = (error) => {
this.setState({isUploading: false});
console.error(error);
}

handleUploadSuccess = (filename) => {
this.setState({avatar: filename, progress: 100, isUploading:
false});
fire.storage().ref('lin_foto').child(filename).getDownloadURL().
then(url => this.setState({avatarURL: url}));
};
```

Purwadhika
Startup and Coding School

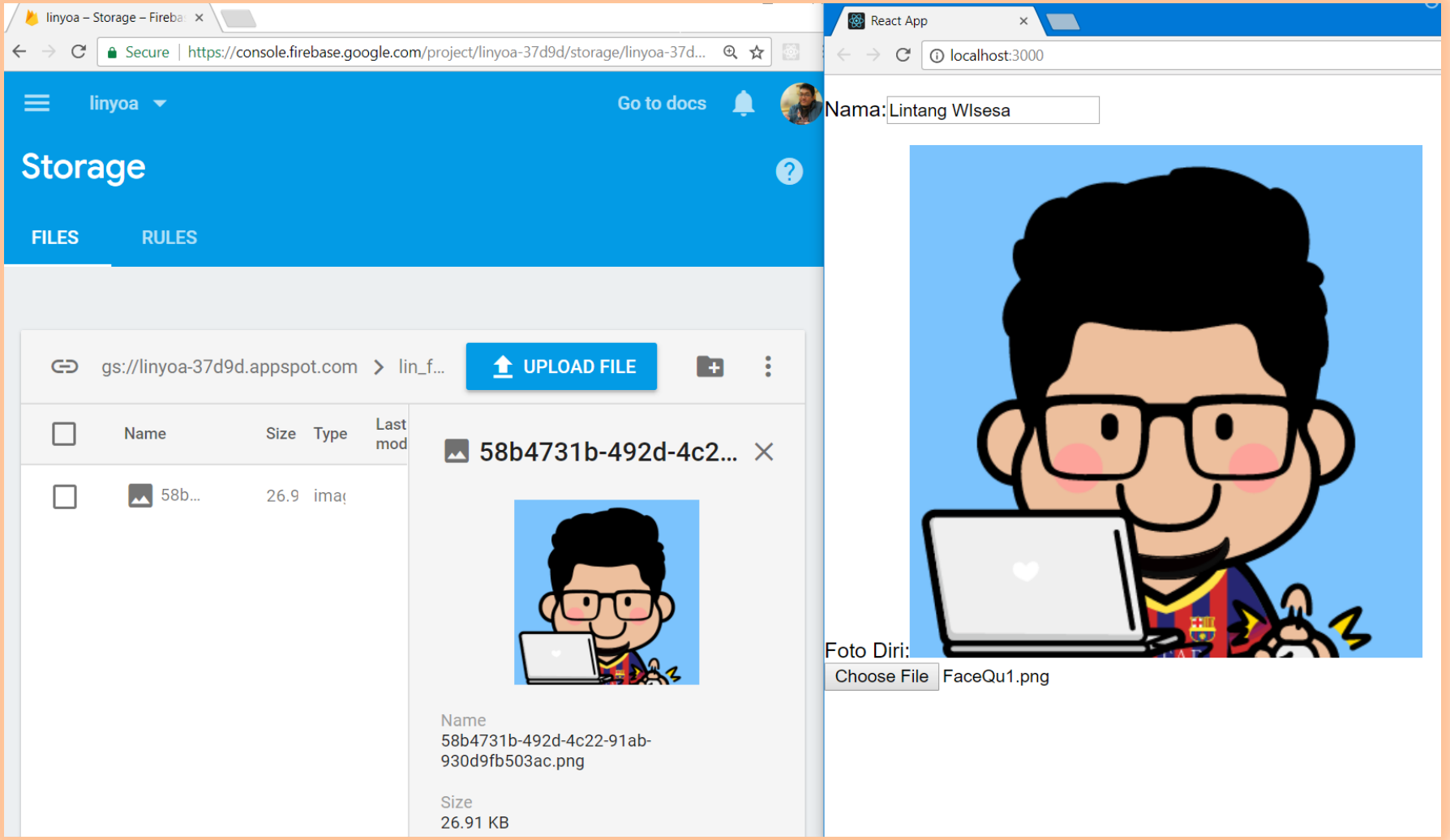
src/App.js

#4 Set Reference to Firebase Storage

```
render() {
return (
<div>
<form>
<p><label>Nama:</label>
<input type="text" value={this.state.username} name="username"
onChange={this.handleChangeUsername} />
</p><label>Foto Diri:</label>
{this.state.isUploading && <p> Process: {this.state.progress}</p>}
{this.state.avatarURL &&
<img src={this.state.avatarURL} alt="ok" />}
<FileUploader accept="image/*" name="avatar" randomizeFilename
storageRef={fire.storage().ref('lin_foto')}
onUploadStart={this.handleUploadStart}
onUploadError={this.handleUploadError}
onUploadSuccess={this.handleUploadSuccess}
onProgress={this.handleProgress}
/>
</form>
</div>
);}}

export default ProfilePage;
```

Purwadhika
Startup and Coding School

React & Firebase Storage



Purwadhika
Startup and Coding School

Front-End Development

React & Firebase

#2 Firebase Database & Storage

Purwadhika
Startup and Coding School