

Front-End Development



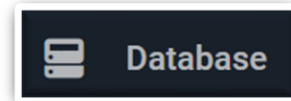
React & Firebase

#2 Firebase Database & Storage

How to Connect React Project to Firebase Database



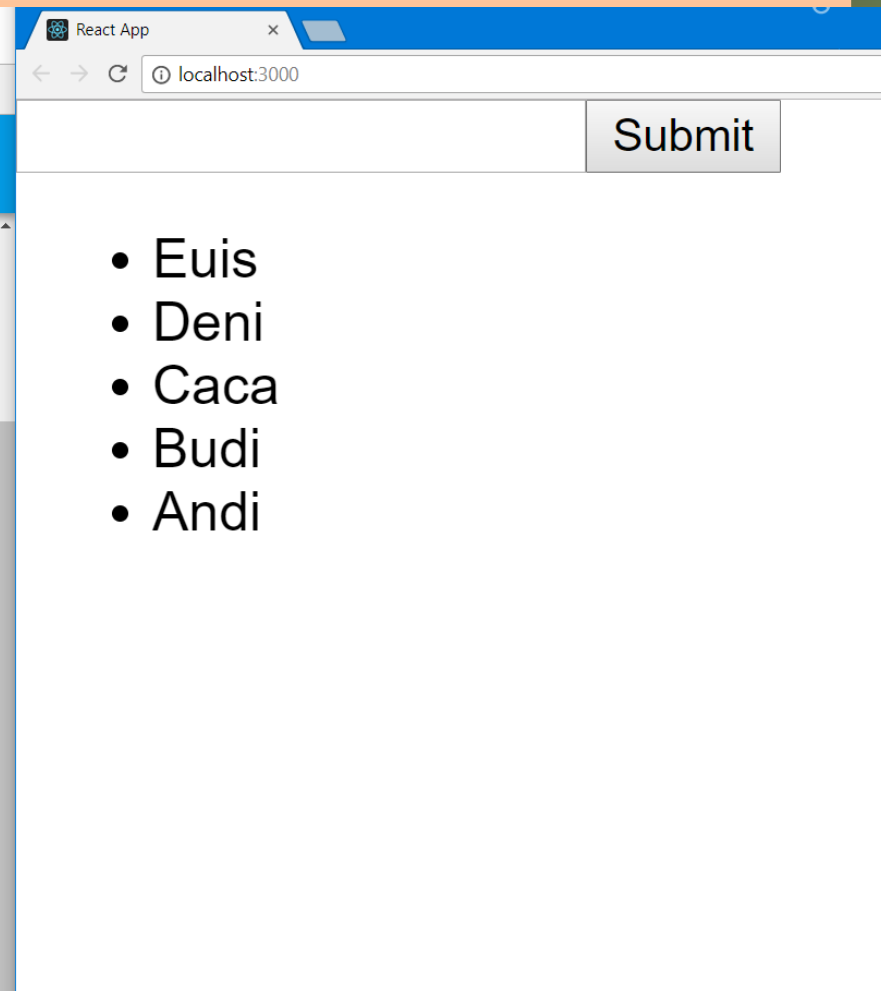
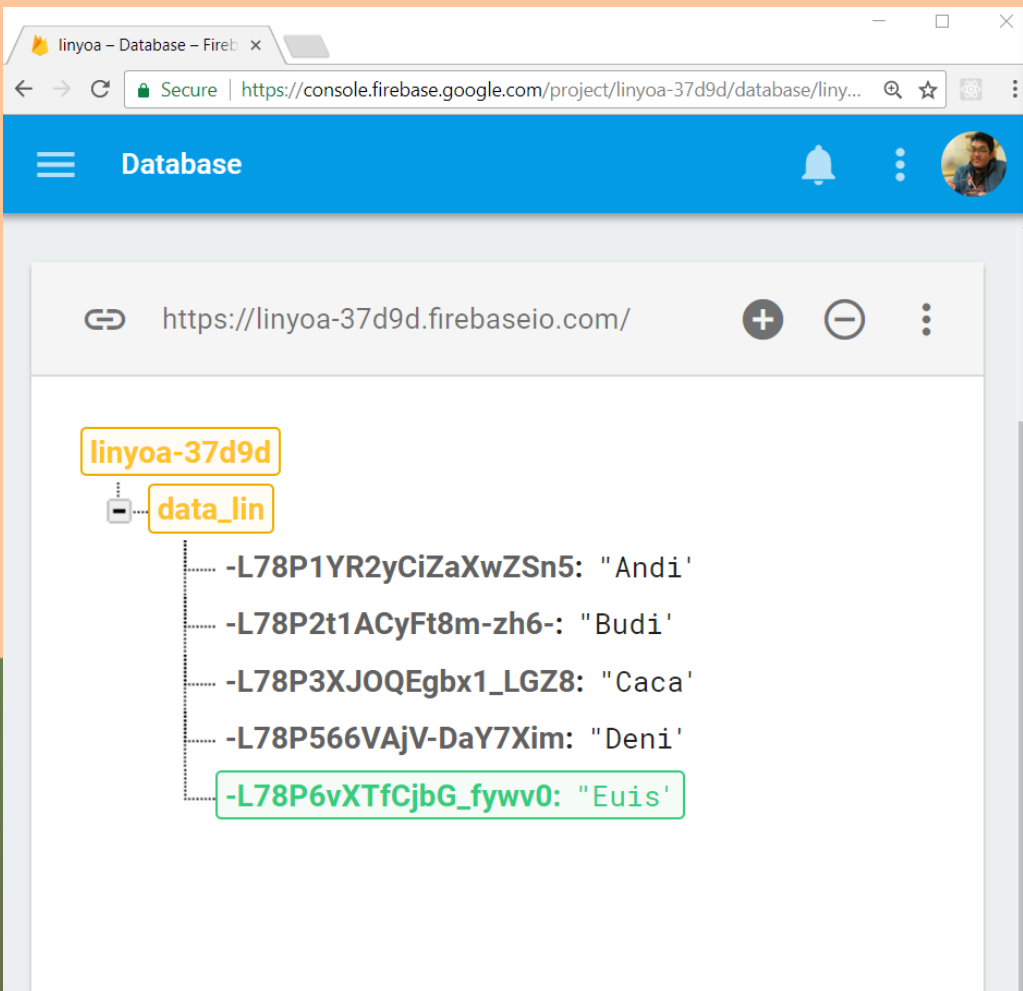
Firebase



Firestore 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.

React & Firebase Database



#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.

Add Firebase to your web app



Copy and paste the snippet below at the bottom of your HTML, before other script tags.

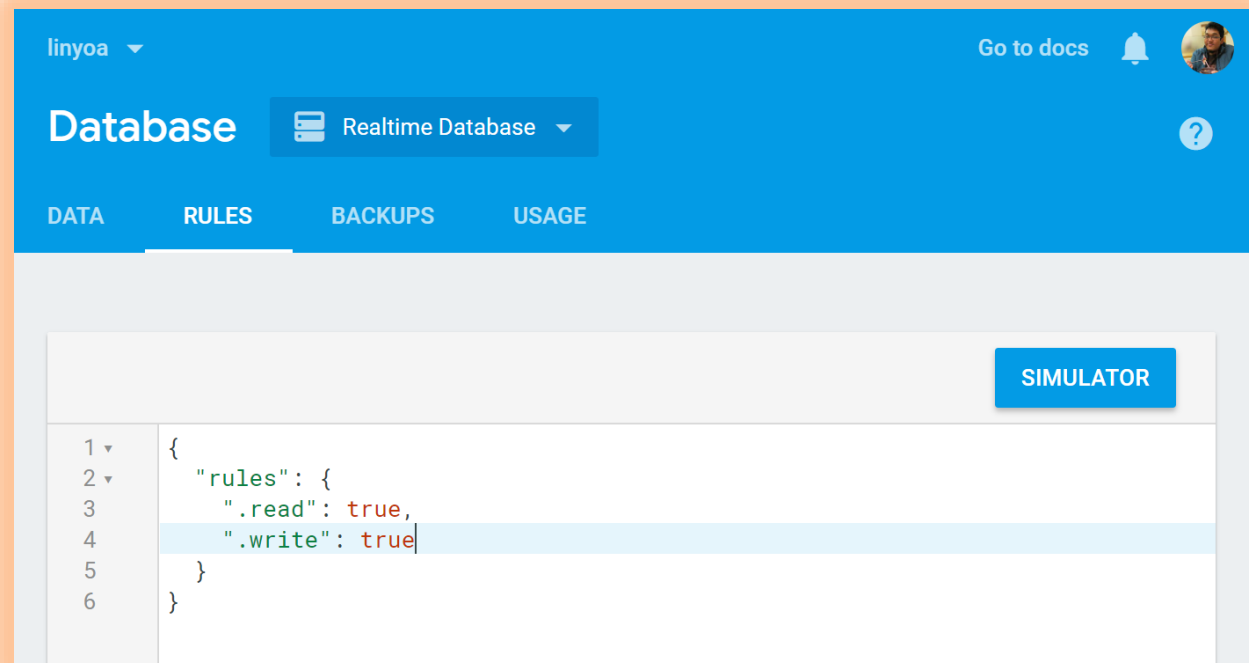
```
<script src="https://www.gstatic.com/firebasejs/4.11.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyAtQf_vMPce7JTfYQesg2isg8pZftJwIII",
    authDomain: "linyoa-37d9d.firebaseio.com",
    databaseURL: "https://linyoa-37d9d.firebaseio.com",
    projectId: "linyoa-37d9d",
    storageBucket: "linyoa-37d9d.appspot.com",
    messagingSenderId: "922933748183"
  };
  firebase.initializeApp(config);
</script>
```

COPY

#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
```

#2 Create/Prepare Your React App

- Install Firebase packages needed:

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

```
$ npm install firebase
```



```
import firebase from 'firebase';

var config = {
  apiKey: "a1b2c3d4e5f6g7h8i9j0",
  authDomain: "linyoa-37d9d.firebaseio.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;
```

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

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

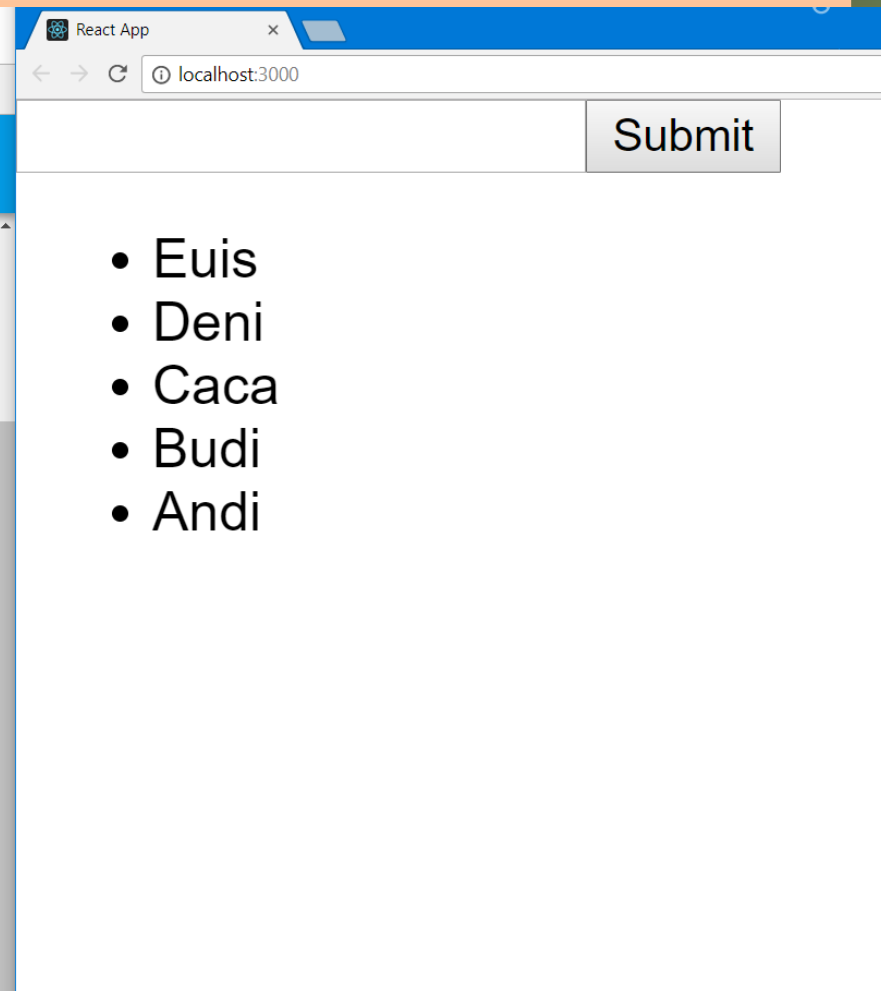
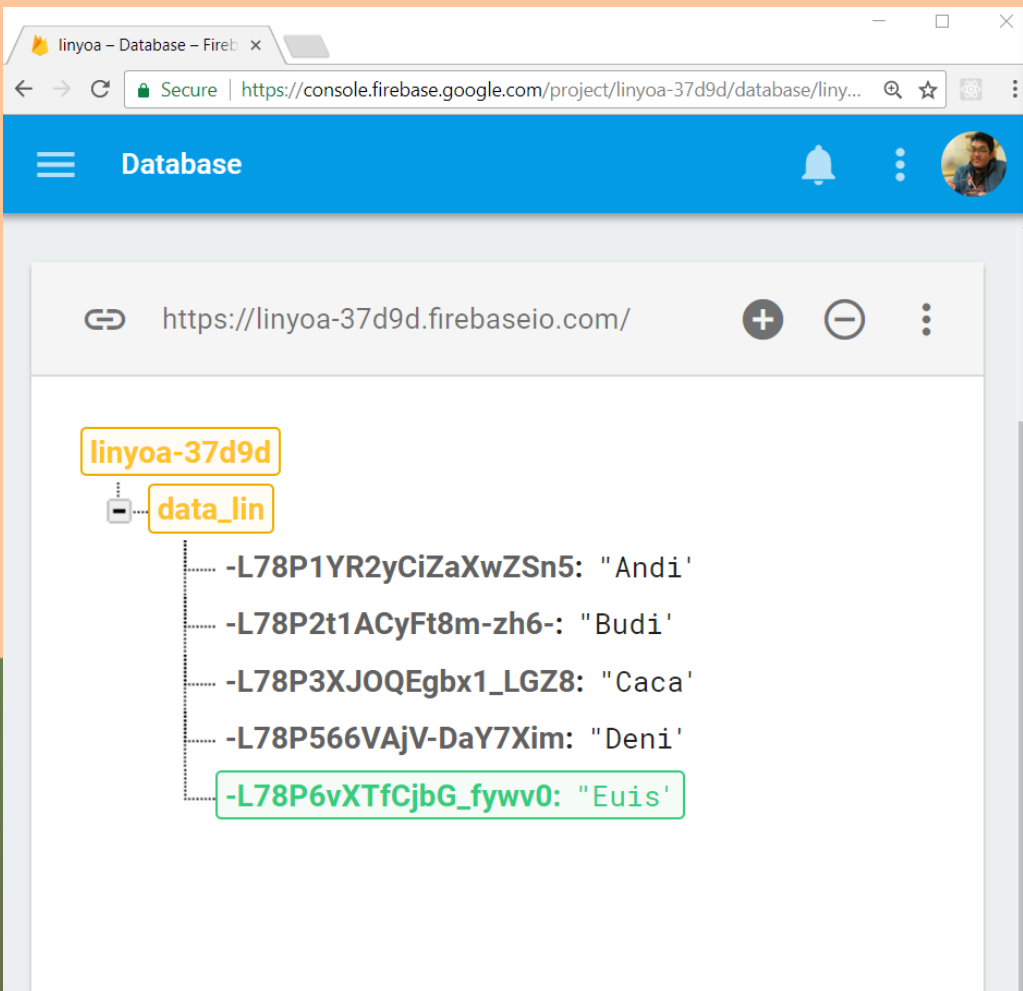
  componentWillMount(){
    let messagesRef =
    fire.database().ref('data_lin').orderByKey().limitToLast(100);
    messagesRef.on('child_added', snapshot => {
      let message = { text: snapshot.val(), id: snapshot.key };
      this.setState({ messages: [message].concat(this.state.messages)
    });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

```
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;
```

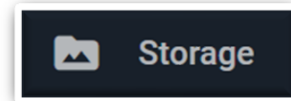
React & Firebase Database



How to Connect React Project to Firebase Storage



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.

React & Firebase Storage

The image shows two side-by-side browser windows. The left window displays the Firebase Storage console for a project named 'linyoa'. The URL is <https://console.firebase.google.com/project/linyoa-37d9d/storage/linyoa-37d...>. The 'Storage' section is active, showing a list of files. A file named '58b4731b-492d-4c2...' is selected, showing a preview of a cartoon character with glasses and a laptop. The file details are: Name: 58b4731b-492d-4c22-91ab-930d9fb503ac.png, Size: 26.91 KB. The right window shows a React application running on localhost:3000. It has a form with a text input labeled 'Nama:' containing the value 'Lintang Wllesa'. Below the form is a large image of the same cartoon character. Below the image is a label 'Foto Diri:' followed by a 'Choose File' button and the filename 'FaceQu1.png'.

linyoa – Storage – Fireba x

Secure | <https://console.firebase.google.com/project/linyoa-37d9d/storage/linyoa-37d...>

linyoa

Go to docs

Storage

FILES RULES

gs://linyoa-37d9d.appspot.com > lin_f... **UPLOAD FILE**

Name	Size	Type	Last mod
58b...	26.9	ima	

58b4731b-492d-4c2... X

Foto Diri:

Choose File FaceQu1.png

#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.

Add Firebase to your web app

×

Copy and paste the snippet below at the bottom of your HTML, before other script tags.

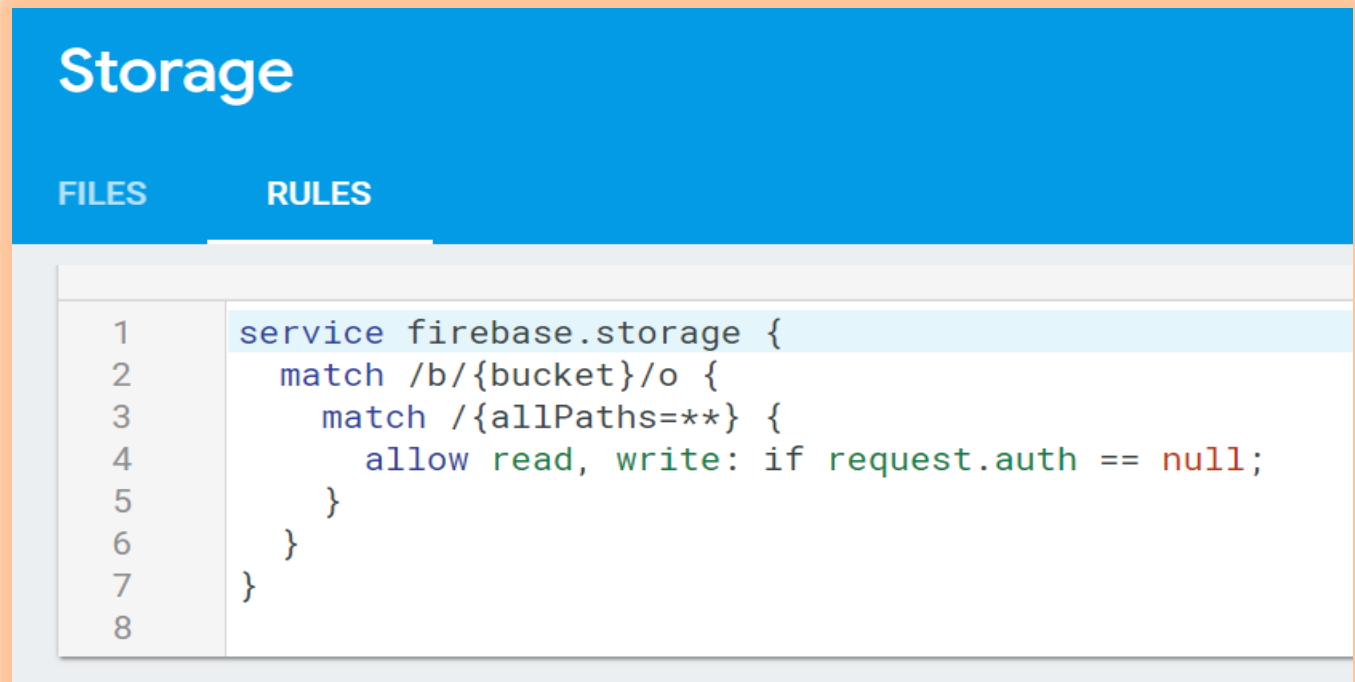
```
<script src="https://www.gstatic.com/firebasejs/4.11.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyAtQf_vMPce7JTfYQesg2isg8pZftJwIII",
    authDomain: "linyoa-37d9d.firebaseio.com",
    databaseURL: "https://linyoa-37d9d.firebaseio.com",
    projectId: "linyoa-37d9d",
    storageBucket: "linyoa-37d9d.appspot.com",
    messagingSenderId: "922933748183"
  };
  firebase.initializeApp(config);
</script>
```

COPY

#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
```

#2 Create/Prepare Your React App

■ Install Firebase packages needed:

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

```
$ npm install firebase
```

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

```
import firebase from 'firebase';

var config = {
  apiKey: "a1b2c3d4e5f6g7h8i9j0",
  authDomain: "linyoa-37d9d.firebaseio.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;
```

```
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: ''
  };
}
```

```
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}));
};
```

```
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;

React & Firebase Storage

The image shows two side-by-side browser windows. The left window displays the Firebase Storage console for a project named 'linyoya'. The URL is <https://console.firebase.google.com/project/linyoya-37d9d/storage/linyoya-37d...>. The interface includes a 'Storage' header, tabs for 'FILES' and 'RULES', and a file list table. The file list contains one entry:

Name	Size	Type	Last mod
58b...	26.9	ima	

Below the table, a preview of the image is shown, along with its name and size:

Name: 58b4731b-492d-4c22-91ab-930d9fb503ac.png
Size: 26.91 KB

The right window shows a React application running on `localhost:3000`. It features a form with a label 'Nama:' and a text input containing 'Lintang Wlsesa'. Below the form is a large image of a cartoon character with glasses and a laptop. Underneath the image, there is a label 'Foto Diri:' and a file selection button labeled 'Choose File' next to the filename 'FaceQu1.png'.

Front-End Development



React & Firebase

#2 Firebase Database & Storage