**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM**



**BÁO CÁO SEMINAR**

**ĐỀ TÀI:**

# React Native cho Truy cập Hệ thống Tập tin và Tải lên/Tải xuống Tập tin

**Giảng viên hướng dẫn: Võ Ngọc Tân**

**Thực hiện bởi nhóm 3, bao gồm:**

1. Trần Vũ Minh 22540011
2. Lê Hữu Quang 22540016

## 1. Thư viện Truy cập Hệ thống Tập tin

### RN File Picker

Thư viện RN File Picker cho phép chọn tệp từ bộ nhớ thiết bị, cần thiết cho các ứng dụng cần tải lên tài liệu, hình ảnh, hoặc video.

#### Cài đặt:

npm install react-native-document-picker

#### Sử dụng:

import DocumentPicker from 'react-native-document-picker';  
  
const selectFile = async () => {  
 try {  
 const file = await DocumentPicker.pickSingle({ type: [DocumentPicker.types.allFiles] });  
 console.log('Selected file:', file);  
 } catch (err) {  
 if (DocumentPicker.isCancel(err)) console.log('User canceled the picker');  
 }  
};

### RN File Viewer

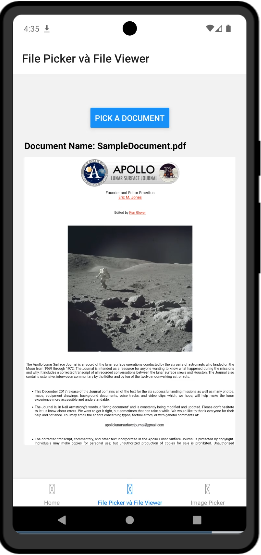
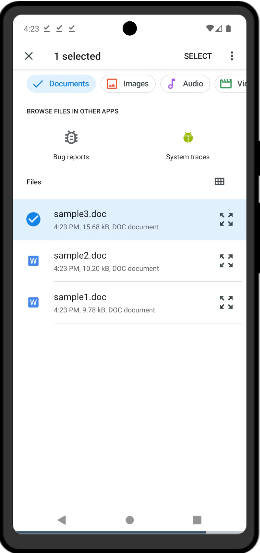
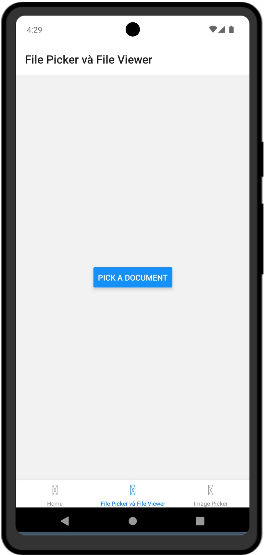
RN File Viewer cho phép xem các tệp trực tiếp trong ứng dụng, không cần chuyển sang trình xem ngoài.

#### Cài đặt:

npm install react-native-file-viewer

#### Sử dụng:

import FileViewer from 'react-native-file-viewer';  
  
const openFile = async (filePath) => {  
 try {  
 await FileViewer.open(filePath);  
 } catch (err) {  
 console.error('Failed to open file:', err);  
 }  
};



### React Native Image Picker

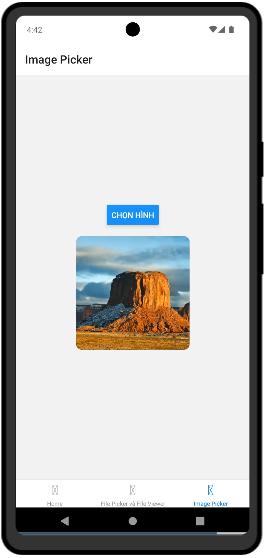
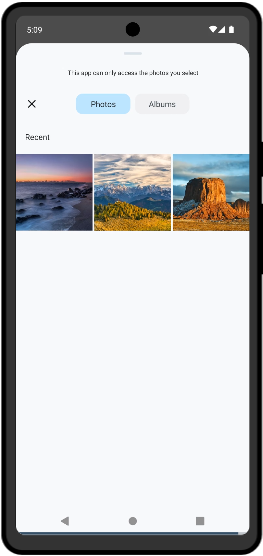
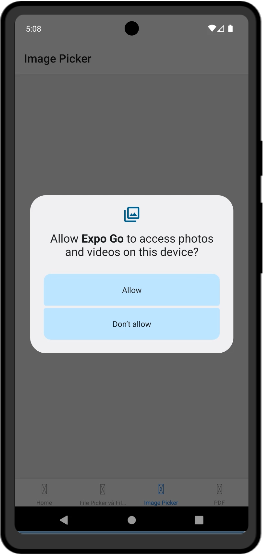
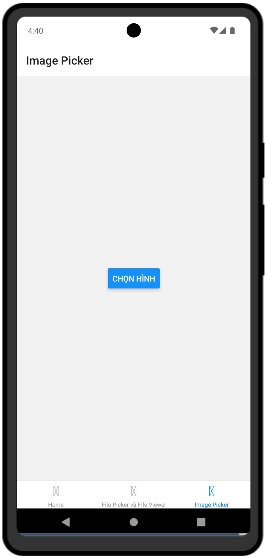
Mô tả: Thư viện React Native Image Picker cho phép người dùng chọn hình ảnh từ thư viện hoặc chụp ảnh trực tiếp, hữu ích cho việc tải hình ảnh lên.

#### Cài đặt:

npm install react-native-image-picker

#### Sử dụng:

var options = {  
 title: 'Select Image',  
 customButtons: [  
 {  
 name: 'customOptionKey',  
 title: 'Choose Photo from Custom Option'  
 },  
 ],  
 storageOptions: {  
 skipBackup: true,  
 path: 'images',  
 },  
};  
ImagePicker.showImagePicker(options, response => {  
 console.log('Response = ', response);  
 if (response.didCancel) {  
 console.log('User cancelled image picker');  
 } else if (response.error) {  
 console.log('ImagePicker Error: ', response.error);  
 } else if (response.customButton) {  
 console.log(  
 'User tapped custom button: ',  
 response.customButton  
 );  
 alert(response.customButton);  
 } else {  
 setFilePath(response);  
 }  
});



### React Native Make PDF

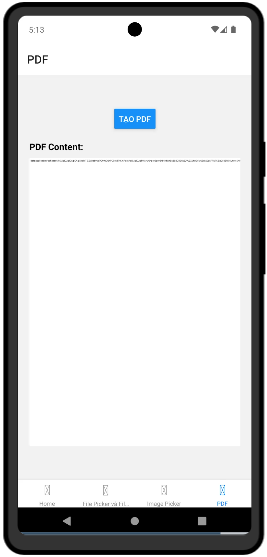
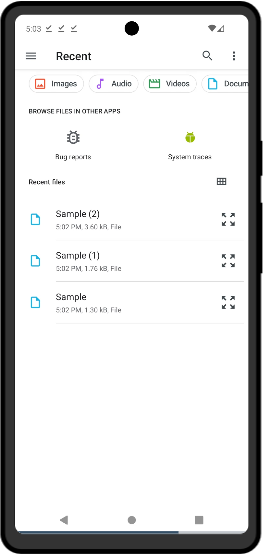
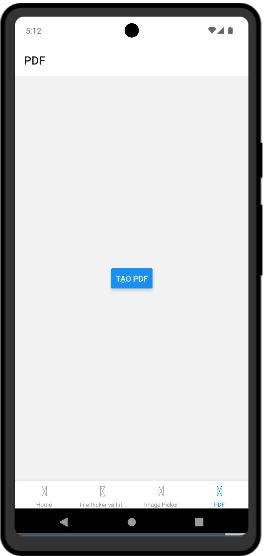
Thư viện này giúp tạo PDF từ nội dung HTML, cung cấp một cách để tạo tài liệu từ dữ liệu động.

#### Cài đặt:

npm install react-native-html-to-pdf --save

#### Sử dụng:

import RNHTMLtoPDF from 'react-native-html-to-pdf';  
  
const App = () => {  
 const [filePath, setFilePath] = useState('');  
  
 const isPermitted = async () => {  
 if (Platform.OS === 'android') {  
 try {  
 const granted = await PermissionsAndroid.request(  
 PermissionsAndroid.PERMISSIONS.WRITE\_EXTERNAL\_STORAGE,  
 {  
 title: 'External Storage Write Permission',  
 message: 'App needs access to Storage data',  
 },  
 );  
 return granted === PermissionsAndroid.RESULTS.GRANTED;  
 } catch (err) {  
 alert('Write permission err', err);  
 return false;  
 }  
 } else {  
 return true;  
 }  
 };  
  
 const createPDF = async () => {  
 if (await isPermitted()) {  
 let options = {  
 //Content to print  
 html:  
 '<h1 style="text-align: center;"><strong>Hello Guys</strong></h1><p style="text-align: center;">Here is an example of pdf Print in React Native</p><p style="text-align: center;"><strong>Team About React</strong></p>',  
 //File Name  
 fileName: 'test',  
 //File directory  
 directory: 'docs',  
 };  
 let file = await RNHTMLtoPDF.convert(options);  
 console.log(file.filePath);  
 setFilePath(file.filePath);  
 }  
 };  
  
 return (  
 <SafeAreaView style={{flex: 1}}>  
 <Text style={styles.titleText}>  
 Example to Make PDF in React Native from HTML Text  
 </Text>  
 <View style={styles.container}>  
 <TouchableOpacity onPress={createPDF}>  
 <View>  
 <Image  
 //We are showing the Image from online  
 source={{  
 uri:  
 'https://raw.githubusercontent.com/AboutReact/sampleresource/master/pdf.png',  
 }}  
 style={styles.imageStyle}  
 />  
 <Text style={styles.textStyle}>Create PDF</Text>  
 </View>  
 </TouchableOpacity>  
 <Text style={styles.textStyle}>{filePath}</Text>  
 </View>  
 </SafeAreaView>  
 );  
};  
  
export default App;



### React Native Print HTML

Thư viện này cho phép in nội dung HTML, hữu ích cho việc in hóa đơn, báo cáo, hoặc các loại tài liệu khác.

#### Cài đặt:

npm install react-native-print

#### Sử dụng:

import RNPrint from 'react-native-print';  
  
const printHTML = async () => {  
 await RNPrint.print({  
 html: '<h1>This is a test print document</h1>'  
 });  
};

## 2. Thư viện Tải lên/Tải xuống Tập tin

### React Native File Upload

Thư viện này cho phép tải tập tin lên máy chủ, một tính năng cơ bản cho các ứng dụng xử lý việc nộp tài liệu.

#### Cài đặt:

npm install react-native-document-picker

#### Sử dụng:

const res = await DocumentPicker.pick({

type: [DocumentPicker.types.allFiles],

});

this.setState({ singleFile: res });

const data = new FormData();

data.append('name', 'Image Upload');

data.append('file\_attachment', fileToUpload);

i let uploadImage = async () => {

//Check if any file is selected or not

if (singleFile != null) {

//If file selected then create FormData

const fileToUpload = singleFile;

const data = new FormData();

data.append('name', 'Image Upload');

data.append('file\_attachment', fileToUpload);

let res = await fetch(

'http://localhost//webservice/user/uploadImage',

{

method: 'post',

body: data,

headers: {

'Content-Type': 'multipart/form-data; ',

},

}

);

let responseJson = await res.json();

if (responseJson.status == 1) {

alert('Upload Successful');

}

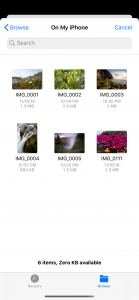
} else {

//if no file selected the show alert

alert('Please Select File first');

}

};



### React Native Download Image

Thư viện này giúp tải xuống hình ảnh, hữu ích cho việc lưu trữ và xem hình ảnh ngoại tuyến.

#### Cài đặt:

npm install react-native-fetch-blob

#### Sử dụng:

const { config, fs } = RNFetchBlob;

let PictureDir = fs.dirs.PictureDir;

let options = {

fileCache: true,

addAndroidDownloads: {

//Related to the Android only

useDownloadManager: true,

notification: true,

path:

PictureDir +

'/image\_' +

Math.floor(date.getTime() +

date.getSeconds() / 2) + ext,

description: 'Image',

},

};

config(options)

.fetch('GET', image\_URL)

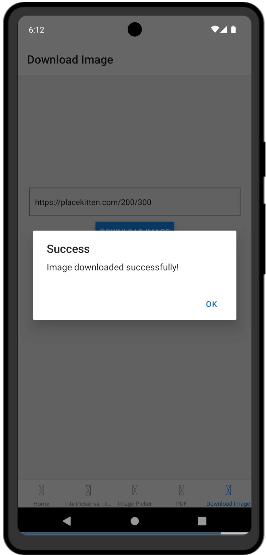
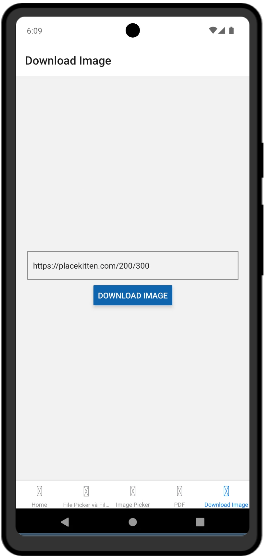
.then(res => {

//Showing alert after successful downloading

console.log('res -> ', JSON.stringify(res));

alert('Image Downloaded Successfully.');

});



### React Native AWS S3 Bucket Upload

Cho phép tải tập tin trực tiếp lên Amazon S3, hữu ích cho các ứng dụng cần lưu trữ an toàn và mở rộng.

#### Cài đặt:

npm install react-native-aws3

#### Sử dụng:

import { RNS3 } from 'react-native-aws3';  
  
const uploadToS3 = async (file) => {  
 const options = {  
 keyPrefix: 'uploads/',  
 bucket: 'your-bucket-name',  
 region: 'us-east-1',  
 accessKey: 'your-access-key',  
 secretKey: 'your-secret-key',  
 };  
 try {  
 const response = await RNS3.put(file, options);  
 console.log('Upload response:', response);  
 } catch (error) {  
 console.error('Upload to S3 failed:', error);  
 }  
};

### React Native Google Drive

Cho phép tương tác với Google Drive, hỗ trợ tải lên và truy xuất tệp.

Cài đặt:

npm install react-native-google-drive-api-wrapper –save

Ngoài ra còn có 1 số thư viện phụ thuộc:

npm install react-native-fs –save

npm install react-native-google-signin --save

Sử dụng:

const \_uploadDriveDataImage = async () => {  
 try {  
 // Check if file selected  
 if (Object.keys(filePath).length == 0)  
 return alert('Please Select any File');  
 setLoading(true);  
 if (!(await \_initGoogleDrive())) {  
 return alert('Failed to Initialize Google Drive');  
 }  
 // Convert Selected File into base64  
 let fileContent = await RNFS.readFile(filePath.uri, 'base64');  
 // console.log('fileContent -> ', JSON.stringify(fileContent));  
 // Create Directory on Google Device  
 let directoryId = await GDrive.files.safeCreateFolder({  
 name: APP\_DIRECTORY,  
 parents: ['root'],  
 });  
 console.log('directoryId -> ', directoryId);  
 // Create Multipart and Upload  
 let result = await GDrive.files.createFileMultipart(  
 fileContent,  
 filePath.type,  
 {  
 parents: [directoryId],  
 name: filePath.name,  
 },  
 true,  
 );  
 // Check upload file response for success  
 if (!result.ok) return alert('Uploading Failed');  
 // Getting the uploaded File Id  
 let fileId = await GDrive.files.getId(  
 filePath.name,  
 [directoryId],  
 filePath.type,  
 false,  
 );  
 alert(`Uploaded Successfull. File Id: ${fileId}`);  
 setFilePath({});  
 } catch (error) {  
 console.log('Error->', error);  
 alert(`Error-> ${error}`);  
 }  
 setLoading(false);  
 };  
  
 const \_uploadDriveData = async () => {  
 try {  
 // Check if file selected  
 if (!inputTextValue) return alert('Please Enter Some Input');  
 setLoading(true);  
 if (!(await \_initGoogleDrive())) {  
 return alert('Failed to Initialize Google Drive');  
 }  
 // Create Directory on Google Device  
 let directoryId = await GDrive.files.safeCreateFolder({  
 name: APP\_DIRECTORY,  
 parents: ['root'],  
 });  
 console.log('directoryId -> ', directoryId);  
 let fileName = new Date().getTime() + '.txt';  
 // Check upload file response for success  
 let result = await GDrive.files.createFileMultipart(  
 inputTextValue,  
 'application/text',  
 {  
 parents: [directoryId],  
 name: fileName,  
 },  
 false,  
 );  
 // Check upload file response for success  
 if (!result.ok) return alert('Uploading Failed');  
 // Getting the uploaded File Id  
 let fileId = await GDrive.files.getId(  
 fileName,  
 [directoryId],  
 'application/text',  
 false,  
 );  
 setInputTextValue('');  
 alert(`Uploaded Successfull. File Id: ${fileId}`);  
 } catch (error) {  
 console.log('Error->', error);  
 alert(`Error-> ${error}`);  
 }  
 setLoading(false);  
 };

