International Journal of Innovative Research in Engineering
Volume 4, Issue 3(May-June 2023), PP: 588-590.
www.theijire.com ***ISSN No: 2582-8746***

Cross Platform Development using React Native

Dikshant Vyas[1], Yogesh Kumar Verma[2], Avisha Dashora[3], Ashwini KumarMali[4], Pradeep Jha[5], Manju Mathur[6]
[1,2,3,4,5,6]*Departmentof Computer Science & Engineering, Global Institute of Technology, Jaipur, India.*

How to cite this paper:
Dikshant Vyas[1], Yogesh Kumar Verma[2], Avisha Dashora[3], Ashwini KumarMali[4], Pradeep Jha[5], Manju Mathur[6], "Cross Platform Development using React Native", IJIRE-V4I03-588-590.

Abstract: *The Research paper studies the Development of cross platform applications using two most popular library and framework, Node.js and React Native. Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows you to run JavaScript on the server side, creating web servers and other server-side applications with JavaScript. React Native is a JavaScript framework for building native mobile applications. It was developed by Face book and allows developers to create mobile applications for iOS and Android using JavaScript and the React library. Some specifications which are explained inside the paper and it also help us to know reason to choose React Native with Node.js and how we can setup both the frameworks in our system and create a project using both the frameworks with database connectivity. Basically, this paper covers everything required for a person or a student to get started with the development of cross platform applications with this wonderful integration and learn the basics to create some beginner to advance level projects related to the MERN stack development and make learning interactive and easy for everyone.*

Key Word: *React Native; Node JS;*

I.INTRODUCTION

Node Js and React Native both are the most popular library and framework. Let's look deeper into each framework one by one:

Node JS

Node.js is a versatile, open-source runtime environment that runs JavaScript code outside of a web browser. It provides developers with the ability to create server-side applications using JavaScript and operate them on a server, enabling the creation of efficient and scalable network applications. It is based on the powerful V8 JavaScript engine, which is the same engine used in Google Chrome.It uses employs an event-driven, non-blocking I/O approach, making it ideal for resource-sensitive, real-time data applications that run across multiple devices. Its lightweight and efficient design make it suitable for handling large amounts of data in real-time.It is based on MVC (Model View Controller) design pattern.

History: Node.js was created in 2009 by Ryan Dahl during his presentation at the European JSConf. Dahl was inspired to create Node.js after seeing a file upload progress bar on Flickr. He wanted to create a way to easily build fast and scalable network applications with JavaScript, and thus, Node.js was born. Initially, Node.js was met with skepticism and was not widely adopted. However, over time, as more and more developers started using it, the Node.js community grew and it became more popular. Today, Node.js is used by companies such as Netflix, Uber, and PayPal for building fast and scalable network applications.

Installation & Creation of new project

1. We need to install Node JS in our local environment meaning node and npm need to be installed.
2. Run: "cd project Name" to navigate into the new project folder..
3. Run: "npminit" and follow the prompts to create a package. json file. This file containsmeta data about your project, such as the project's name, version, and dependencies.

After completing these steps, you should have a new Node.js project set up and ready to go. You can then start installing packages and dependencies for your project using npm, and writing your Node.js code.

MVC Structure of Node JS

In a Node.js application, each of MVC (Model, View, Controller) components can be implemented in a variety of ways. There are many popular libraries and frameworks available for building MVC applications with Node.js, such as Express, Hapi, and Loopback. These libraries provide a set of tools and conventions for structuring an MVC application and make it easier to build scalable and maintainable applications.

React Native

React Native is a JavaScript-based platform for creating native mobile applications. It allows you to use React to build native iOS and Android applications. React Native utilizes the same design principles as React, allowing you to build a dynamic mobile UI using declarative components.

React Native enables you to develop mobile apps that have the same look and feel as apps built with Objective-C or Java. The framework uses the same UI elements as native iOS and Android apps and allows you to assemble them using JavaScript and the React library.

One of the main benefits of React Native is that it allows you to build applications that run natively on both iOS and Android, using a single codebase. This can save a significant amount of time and effort for developers, as it eliminates the need to build separate versions of the app for each platform.

React Environment Setup

- To set up a new React Native project, you will need to install the React Native command-line interface (CLI) globally on your machine. You can do this by running the following command in your terminal: ` npm install -g react-native-cli`
- Once the React Native CLI is installed, you can create a new project by running the following command: ` react-native init My Project`
- This will create a new directory called My Project that contains the skeleton of a React Native project. You can then navigate into the project directory by running: ` cd My Project`
- To run your React Native app on an Android emulator or device, you will need to have Android Studio and the Android SDK installed on your machine. You can then launch the emulator and run the following command in your project directory: ` react-native run-android`
- To run your app on an iOS simulator, you will need to have X code installed on your machine. You can then launch the simulator and run the following command in your project directory: ` react-native run-ios`

Advantages of connecting react native with node js: There are a few reasons why you might want to use Node.js with React Native:

- You can use Node.js as a server-side runtime environment for your React Native app. This can be useful if you want to build a back-end service that your app can communicate with over a network.
- You can use Node.js to build command-line tools that can be used to automate tasks related to your React Native app development process, such as building and deploying your app.
- You can use Node.js to build custom scripts that can be used to extend the functionality of your React Native app or integrate it with other tools and services.
- You can use Node.js to build and run tests for your React Native app, using tools like Jest and Enzyme.
- Overall, using Node.js with React Native can provide a number of benefits and can help you build powerful and scalable apps.

II.ABOUT THE PROJECT

In this project we are going to build a website and a cross platform mobile application using MERN Stack and React native. It is a user-friendly website and app which helps user to find suitable car for them by providing information including following specifications:

- Price
- Engine Specs
- Mileage
- Safety Rating
- Features

The user also gets the option to set appropriate filters as per their requirement. Our application also supports the ease to compare different cars based on above mentioned specifications.

III.SETTING UP THE BACKEND

Create a project folder for Node JS by running a command `npminit -y`. You must have latest version of NodeJS and NPM installed.

Step 1: Install Express Package

npm install express –save

Step 2: Install nodemon(dev dependency)

npm install nodemon –save-dev

Step 3: Install below packages for project

•mongoose
•cors
•morgan
•dotenv

Step 4: Create a server.js file
• initialize express
• setup server port number

Step5: Create package script
• add `npm start` script to package.json
• set npm start as `nodemon server.js`.

IV.SETTING UP THE FRONTEND

Create Use the following commands to get ready for React Native Application. Use Bootstrap, Ant Design and Material UI for styling and better look and feel.

Step 1:npx create-react-native-app myapp

Step 2: cd myapp

Step 3:npm install bootstrap @mui/material antd

Step 4:npmaxios

Axios is the main tool for connecting back-end with front-end. All the requests will be sent to the server (backend) with the help of Axios.

V.SETTING UP THE DATABASE

We are using Mongo DB as a backend database for our Project. The best part is that we can setup Node JS with Mongo DB by adding just one line of code. There is no need to change controller, views, or any other modules.

Working: Mongo DB is a No SQL database solution that uses semi-structured data represented in documents, similar to JSON. It is favored among web developers as it can effortlessly be paired with Node.js applications through the Mongo DB driver.

Features:

1. Scalability- Mongo DB is designed to scale horizontally, meaning you can easily add more servers to your system as your database grows.
2. Flexibility- Mongo DB adopts a document-oriented approach to data storage, allowing you to save information in the form of BSON (Binary JSON), which is a flexible, JSON-like document format.
3. This allows you to easily add or change fields in your documents without having to update the entire database schema.
4. Performance- Mongo DB is known for its high performance and ability to handle large amounts of data. It uses a memory-mapped file system and a multi-threaded architecture to achieve fast read and write speeds.
5. Easy to use- Mongo DB has a simple and intuitive API that makes it easy for developers to work with. It also has a large and active community, which means you can find plenty of resources and support online.
6. Cloud-native- Mongo DB is designed to be deployed in the cloud and offers a number of features that make it well-suited for cloud-based applications, including automatic sharding and horizontal scaling.

VI.CONCLUSION

In conclusion, this paper has explored the topic of cross-platform development using React Native and has shed light on its significance and benefits in today's software development landscape. React Native has emerged as a powerful framework that enables developers to create mobile applications for multiple platforms, including iOS and Android, using a single codebase.

References

[1]. "Learning React Native: Building Native Mobile Apps with JavaScript 2nd Edition" by Bonnie Eisenman.
[2]. "React Native in Action - Developing iOS and Android apps with JavaScript " by Nader Dabit.
[3]. The official React Native documentation (https://reactnative.dev/).
[4]. "Node.js in Action 2nd Edition" by Alex R. Young, Bradley Meck, Mike Cantelon, Tim Oxley, Marc Hater, TJ Holowaychuk, and Nathan Rajlich.
[5]. "Node.js the Right Way: Practical, Server-Side JavaScript That Scales 1st Edition" by Jim R. Wilson.
[6]. "The Road to React: Your journey to master React.js in JavaScript" by Robin Wieruch.
[7]. The Node.js tutorials on the MDN Web Docs (https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs).
[8]. "React in Action" by Mark Tielens Thomas.
[9]. "MongoDB: The Definitive Guide" by Kristina Chodorow and Michael Dirolf.
[10]. "Practical MongoDB: Architecting, Developing, and Administering MongoDB First Edition" by Shakuntala Gupta Edward and Navin Sabharwal.
[11]. The MongoDB University (https://university.mongodb.com/).