



**SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
MAHAGAON**

A

PROJECT REPORT

On

ONLINE CAR PARKING APPLICATION USING GPS MAPPING

**SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD
OF THE DEGREE**

BACHELOR OF TECHNOLOGY

(Computer Science and Engineering)

SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Name of Student	Roll No
Mr. Tanishq Ashok Injal (Leader)	20
Ms. Ankita Ashok Madakari	22
Ms. Shivani Nandkumar Nimbalkar	23
Ms. Shweta Vilas Jadhav	21

GUIDED BY

Mrs. P. G. Sanmane

Academic Year
2021-2022



CERTIFICATE

This is to certify that the project entitled “**ONLINE CAR PARKING APPLICATION USING GPS MAPPING**”, which is being submitted by,

Mr. Tanishq Ashok Injal	(20)
Ms. Ankita Ashok Madakari	(22)
Ms. Shivani Nandkumar Nimbalkar	(23)
Ms. Shweta Vilas Jadhav	(21)

for the ‘Bachelor of Computer Science and Engineering’ of **Shivaji University, Kolhapur**. This is the bonafide work carried under my supervision and guidance.

Place:

Date:

Prof. P. G. Sanmane
Guide
Department of CSE

Prof. S. S. Gurav
Project Coordinator
Department of CSE

Prof. S. G. Swami
H.O.D
Department of CSE

Dr. S. H. Sawant
Principal
SGMCOE

UNDERTAKING

We hereby confirm that the work which is being presented in the final year Project Report entitled “**ONLINE CAR PARKING APPLICATION USING GPS MAPPING**”, in the partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** and submitted to the Department of Computer Science and Engineering of Sant Gajanan Maharaj College of Engineering, Mahagoan, Maharashtra is an authentic record of my own work carried out during a period from **Sept 2021 to Jan 2022(7th semester)** under the supervision of **Prof. Mrs. P. G. Sanmane, Assistant Professor, CSE Department.**

The matter presented in this Project Report has not been submitted by any one of us for the award of any other degree elsewhere.

Signature of Student (S)

Mr. Tanishq Ashok Injal (20)	_____
Ms. Ankita Ashok Madakari (22)	_____
Mr. Shivani Nandkumar Nimbalkar (23)	_____
Ms. Shweta Vilas Jadhav (21)	_____

ACKNOWLEDGEMENT

We would like to express our gratitude to world respected Hon. Founder Chairman **Adv. Annasaheb D. Chavan**, all Board of Directors, our beloved Principal **Dr. S. H. Sawant** for their encouragement and support.

We would like to express our gratitude to words respected guide **Mrs. P. G. Sanmane** for her constant encouragement and valuable guidance during the completion of this project.

We are very thankful to **Prof. S. G. Swami (HOD)** for his valuable co-operation & guidance during this project work.

We take this opportunity to thank the entire staff member for their co-operation and their helpfulness during this project work.

Last but not the least assistance offered by various friends and colleagues related directly or indirectly to this work is also gratefully acknowledged.

Mr. Tanishq Ashok Injal

Ms. Ankita Ashok Madakari

Ms. Shivani Nandkumar Nimbalkar

Ms. Shweta Vilas Jadhav

ABSTRACT

Online car parking application using GPS mapping is an application built to book parking slots by allotting free parking slots. This application is to reduce the traffic in parking slots. Normally we can see in the multiplexes, cinema halls, large industries, shopping-malls, and function halls there is problem, people have to go and search which line is empty and which line having slots to park the vehicle. Hence for parking, there is need of workers for parking vehicles in correct position and it's a money consuming process.

Also, when people park their vehicles in no parking zones their vehicles are taken away by towing vans and they have to pay the penalty. This problem can be solved by providing people a platform to easily book their parking slots before going on roads with this online parking slots booking application.

LIST OF FIGURES

Fig. No.	Title
1	System Architecture
2	Data flow Diagram
3	Flowchart

LIST OF ABBREVIATIONS

Abbreviation	Full Form
API	Application Programming Interface
DB	Database
NPM	Node Package Manager
URI	Uniform Resource Identifier

CONTENTS

Acknowledgement	4
Abstract	5
List of Figures	6
List of Abbreviations	6
1. INTRODUCTION	8
1.1 Introduction	8
1.2 Literature Survey	8
1.3 Need of Present Work	8
2. PROBLEM STATEMENT	9
2.1 Problem statement	9
2.1.1 Scope	9
3. SYSTEM DESIGN	10
3.1 System Architecture	10
3.2 Modules	11
3.3 Data Flow Diagram	12
3.4 Flowchart	13
4. IMPLEMENTATION AND EXPERIMENTAL SETUP	14
4.1 Software Tool	14
5.EXPERIMENTAL RESULT AND ITS ANALYSIS	15
5.1 Implementation Results	15
5.2 Environment set up and configuration	19
6. REFERENCES	28
6.1 Web References	28
6.2 IEEE Paper References	28

1. INTRODUCTION

1.1. INTRODUCTION:

The smart parking application aims to help users to find slots for parking, make bookings and extend time if required. User access location-based information and request system services via mobile. Due to increasing population and limited land, cities face huge parking problems almost every day. People just park their vehicles wherever they find a free space. This leads to illegal parking where the owner of those premises is unaware of such activities. Also, in parking areas like super market's parking where there is huge crowd, face parking problems which leads to deadlocks i.e., where vehicles wait for parking areas to get empty and parking slots are already occupied.

As parking now-a-days produce various issues, it creates a necessity to introduce an automated system that allows users to book their parking slots just by making few clicks. With this new technology this has been made possible with the help of mobile app that will help you to locate a free parking slot. This application will help to reduce the searching time for parking the vehicles. People will first book the slots and then go on roads on time. This will also help to reduce some amount of traffic on roads. Illegal parking on roads will be reduced. Anytime anywhere booking will be available with the help of this application.

1.2 Literature Survey:

- a) Ashok, D., Tiwari, A., & Jirge, V. (2020). *Smart Parking System using IoT Technology*. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). doi:10.1109/ic-etite47903.2020.457
- b) Mohammadi, F., Nazri, G.-A., & Saif, M. (2019). *A Real-Time Cloud-Based Intelligent Car Parking System for Smart Cities*. 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP). doi:10.1109/icicsp48821.2019.8958
- c) Patil, B. K., Deshpande, A., Suryavanshi, S., Magdum, R., & Manjunath, B. (2018). *Smart Parking System for Cars*. 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE). doi:10.1109/icrieece44171.2018.90

1.3 Need of Present Work:

The proposed project is a smart parking booking application that provides customers an easy way of reserving a parking space online. It overcomes the problem of finding a parking space in commercial areas that unnecessary consumes time. Hence this project offers an android-based reservation application where users can view various parking areas and select the space to view whether space is available or not.

2. PROBLEM STATEMENT

2.1 PROBLEM STATEMENT:

Time wasted due to inconvenient parking slots and accidents caused by disorganized parking slots leading to more CO2 emission creates a necessity to introduce an automated system that allows users to book their parking slots just by making few clicks.

2.1.1 Scope:

Using the slot allocation method, we can book our parking slot. It is an efficient one for solving parking problems, which overcomes the traffic congestion also provides automated billing process. People will first book the slots and then go on roads on time. This will also help to reduce some amount of traffic on roads. Illegal parking on roads will be reduced. Anytime anywhere booking will be available with the help of this application.

3.SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE –

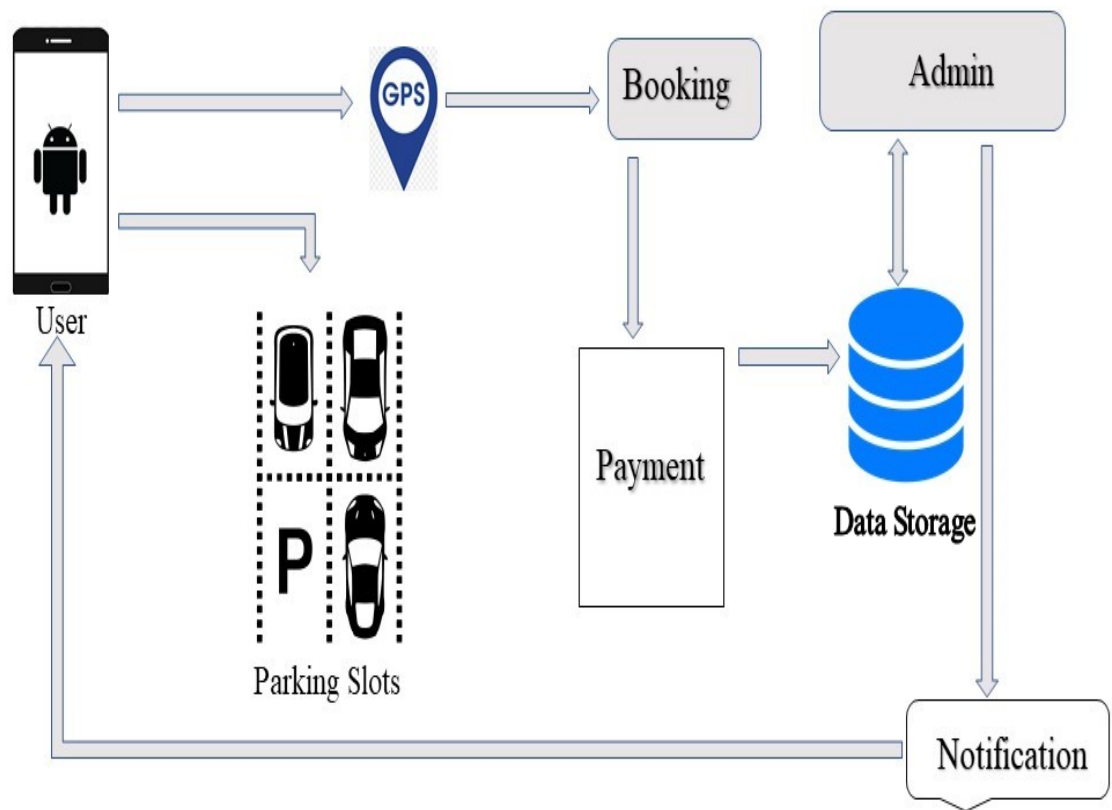


Fig. System Architecture of Online Car Parking Application Using GPS Mapping

3.2 Modules:

User:

User will register and login in the application and find the empty slots in required location with the help of GPS.

GPS:

GPS helps the user to find the suitable parking slots and book the slots by showing appropriate locations of a specific area.

Booking:

After selecting the slot, the user books the slots as per their requirement.

Payment:

After booking the user can pay online the parking fees through payment gateway.

Data storage:

All the data of user with their booking and payment details will be saved in the database which will be further accessed by admin and act accordingly.

Admin:

Admin can access the user data from data storage and will be able to manage that data and trigger notification.

Notification:

Notification module is been handled by admin. The admin sends the notification of booking payment and time will be sent to the user.

3.3 Data Flow Diagram:

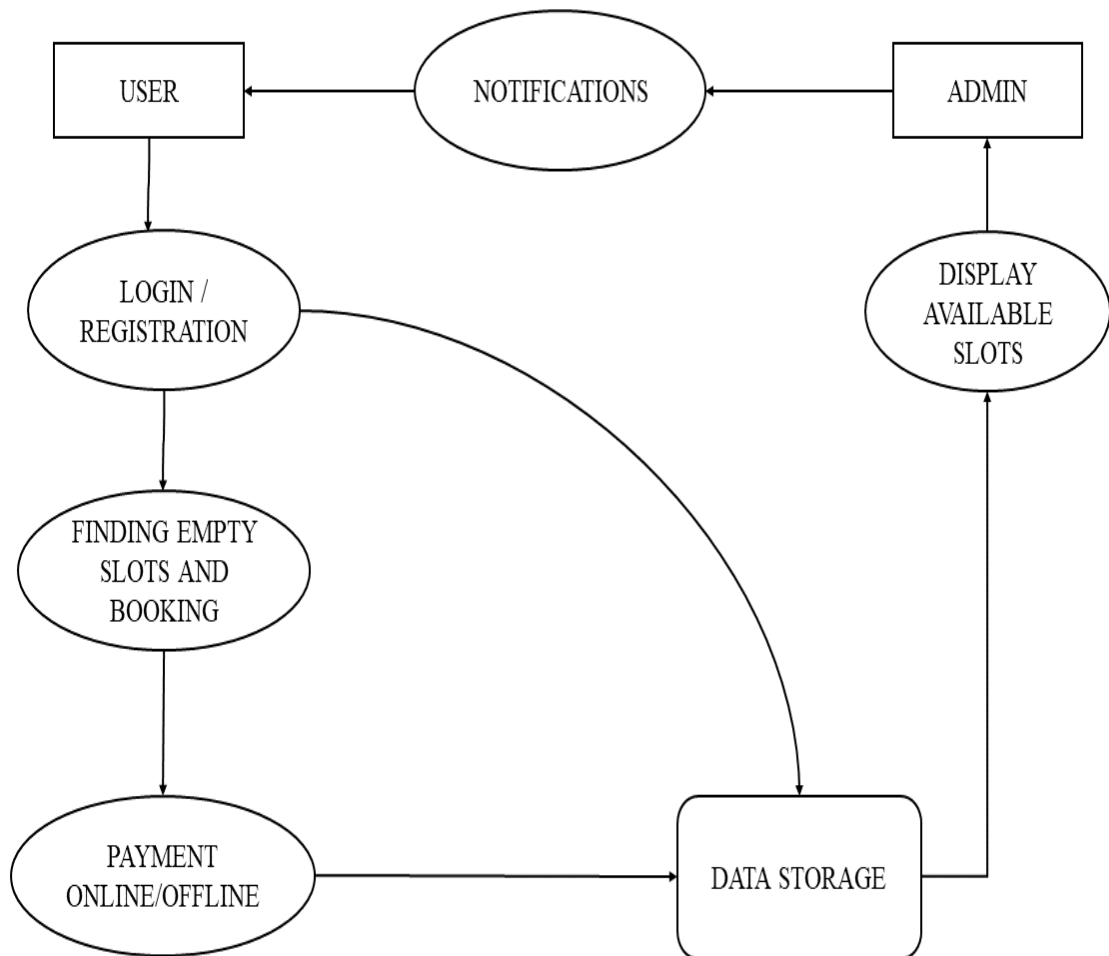


Fig. Data Flow Diagram of Online Car Parking Application Using GPS Mapping

In the above data flow diagram how the data flows in this application. First the user logs in or register to application. That data is stored in data storage and further the user goes to find empty slots and book their parking slot. After selecting the required parking slot user is redirected to payment gateway and its data is given to data storage. According to the data from data storage admin gets all data information and accordingly sends notifications to user.

3.4 Flowchart:

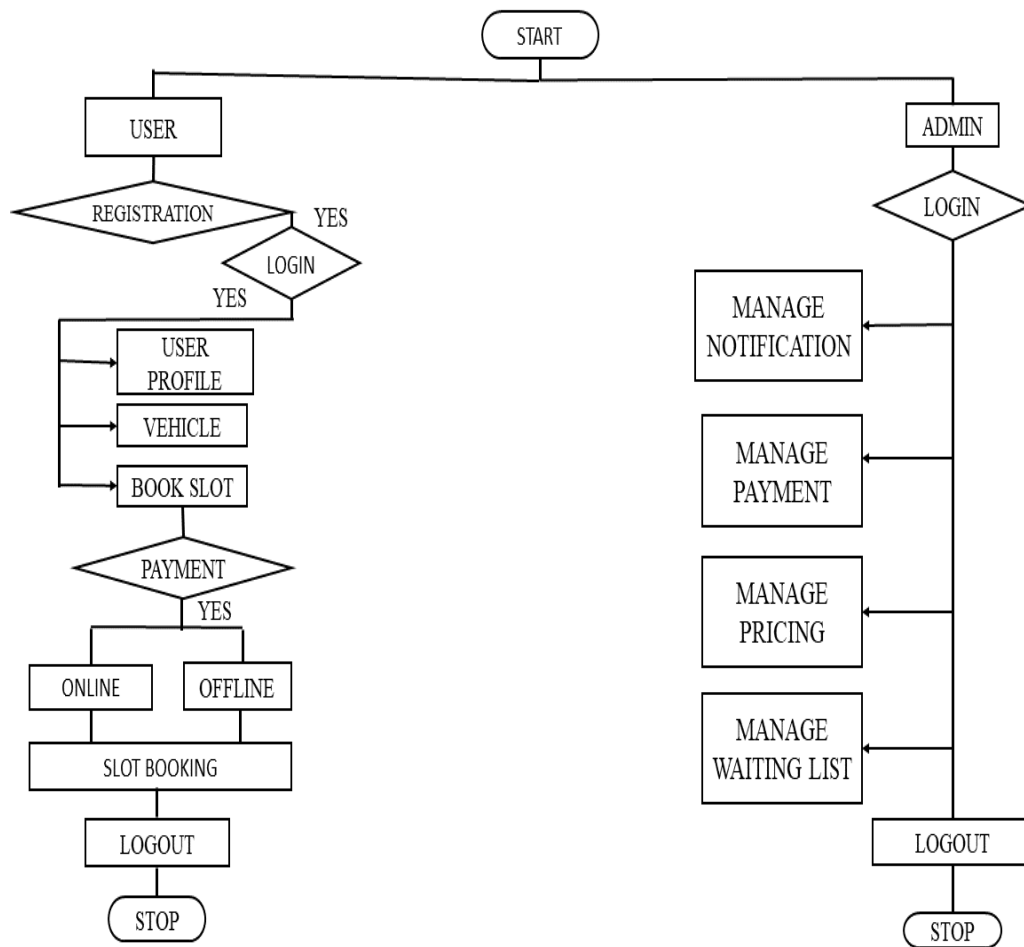


Fig. Flowchart of Online Car Parking Application Using GPS Mapping

4. IMPLEMENTATION AND EXPERIMENTAL SETUP:

4.1 SOFTWARE TOOLS:

Android Studio and VS code for frontend.

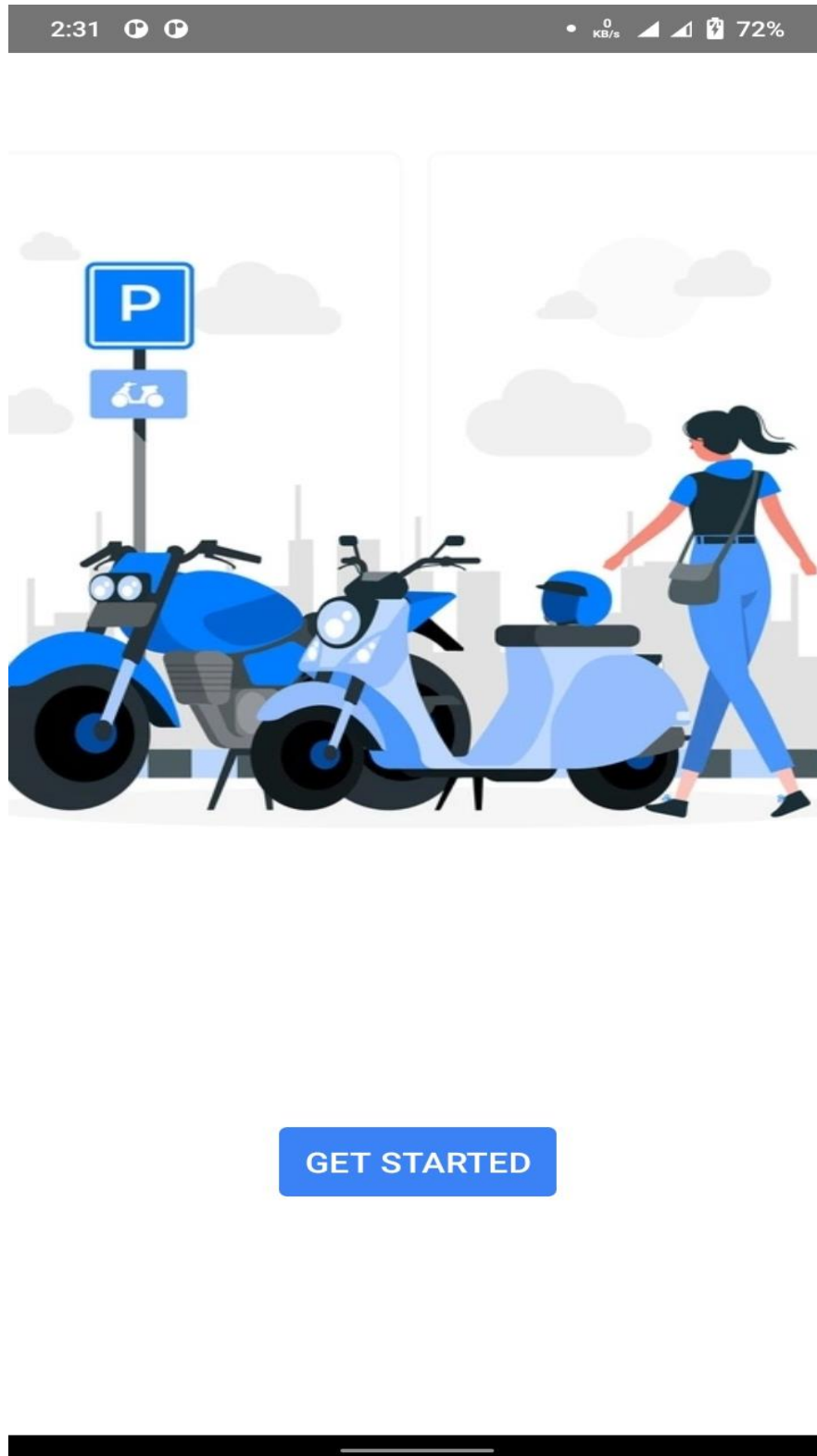
Node JS for backend.

MongoDB for database.


5. EXPERIMENTAL RESULT AND ITS ANALYSIS:

5.1 Implementation Results:

Introductory Screen:



Login Screen:



2:31 0 KB/s 72%

Login

Email ID

e.g:xyz@example.com

Password

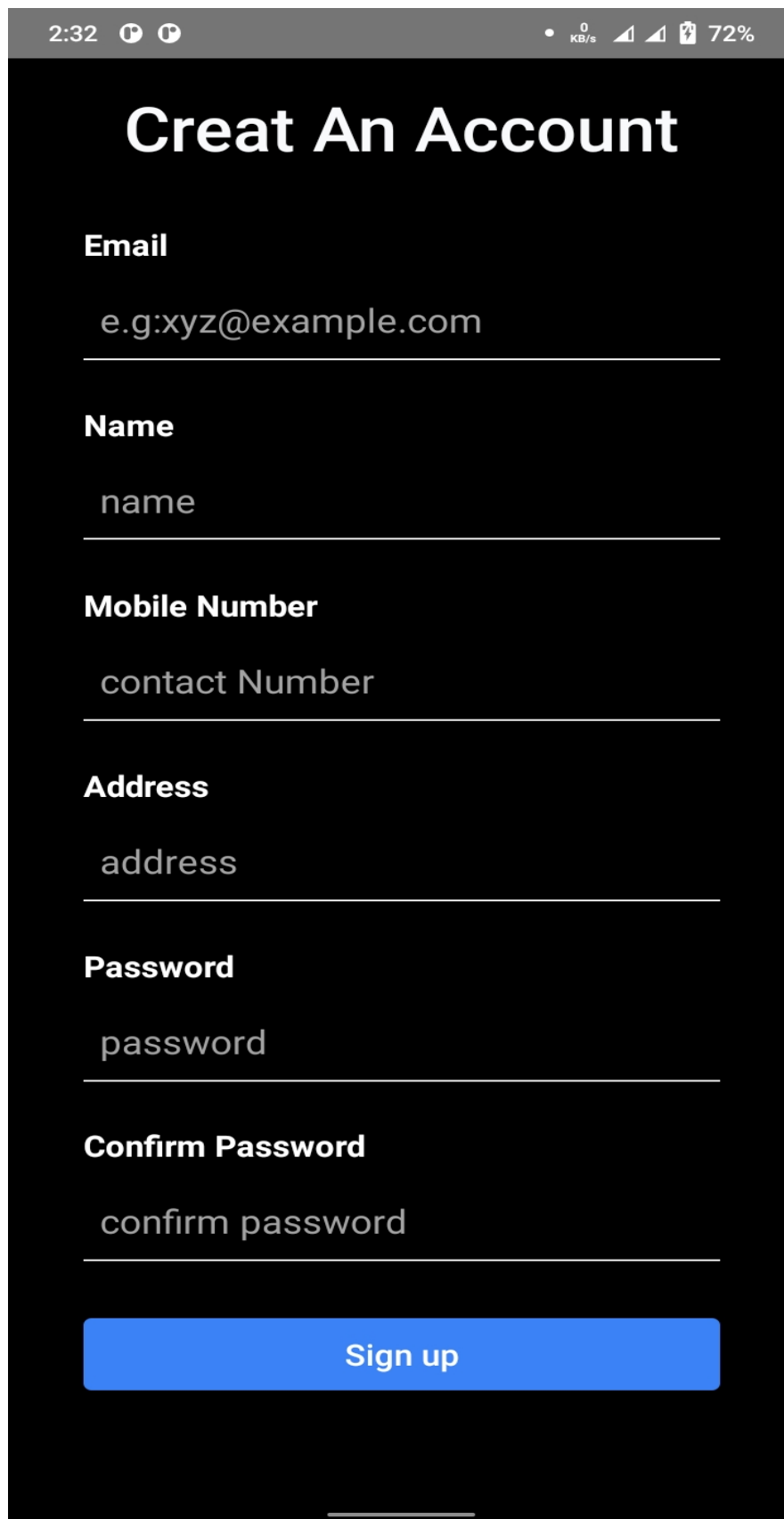
password

Sign in

[Forget Password?](#)

Don't have an account **sign Up**

SignUp / Registration Screen:

A mobile app registration screen with a dark blue background. At the top is a status bar with the time 2:32, signal strength, and battery level at 72%. The main heading is 'Creat An Account' in large white text. Below it are six input fields, each with a label in bold white text and a placeholder in light gray text. The fields are for Email (placeholder: e.g:xyz@example.com), Name (placeholder: name), Mobile Number (placeholder: contact Number), Address (placeholder: address), Password (placeholder: password), and Confirm Password (placeholder: confirm password). At the bottom is a large orange button with the text 'Sign up' in white. A thin white line is visible at the very bottom of the screen.

2:32 0 KB/s 72%

Creat An Account

Email
e.g:xyz@example.com

Name
name

Mobile Number
contact Number

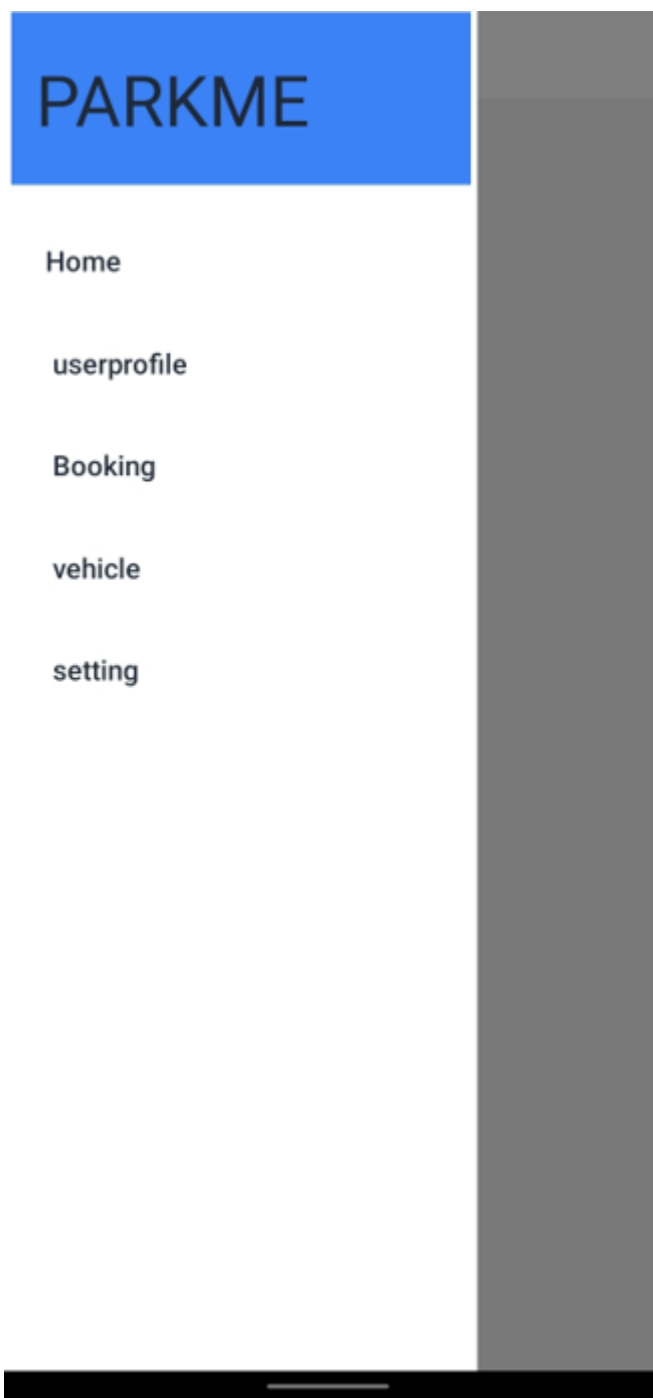
Address
address

Password
password

Confirm Password
confirm password

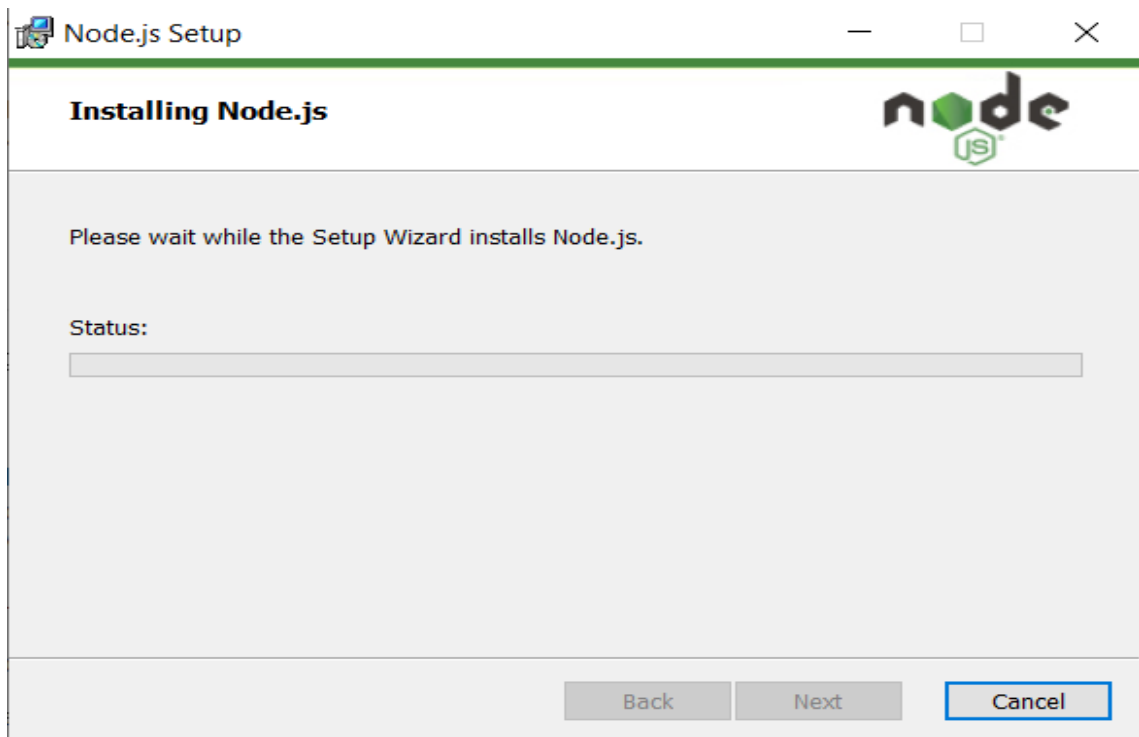
Sign up

Drawer Navigation:

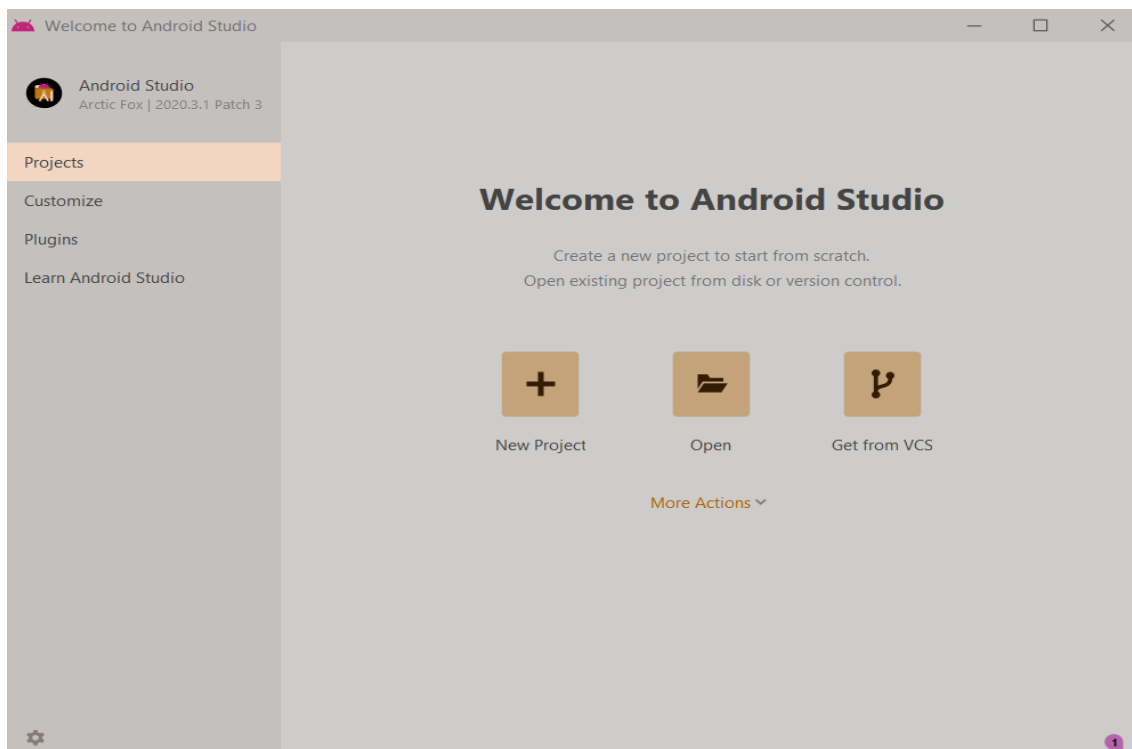


5.2 ENVIRONMENTAL SETUP AND CONFIGURATION:

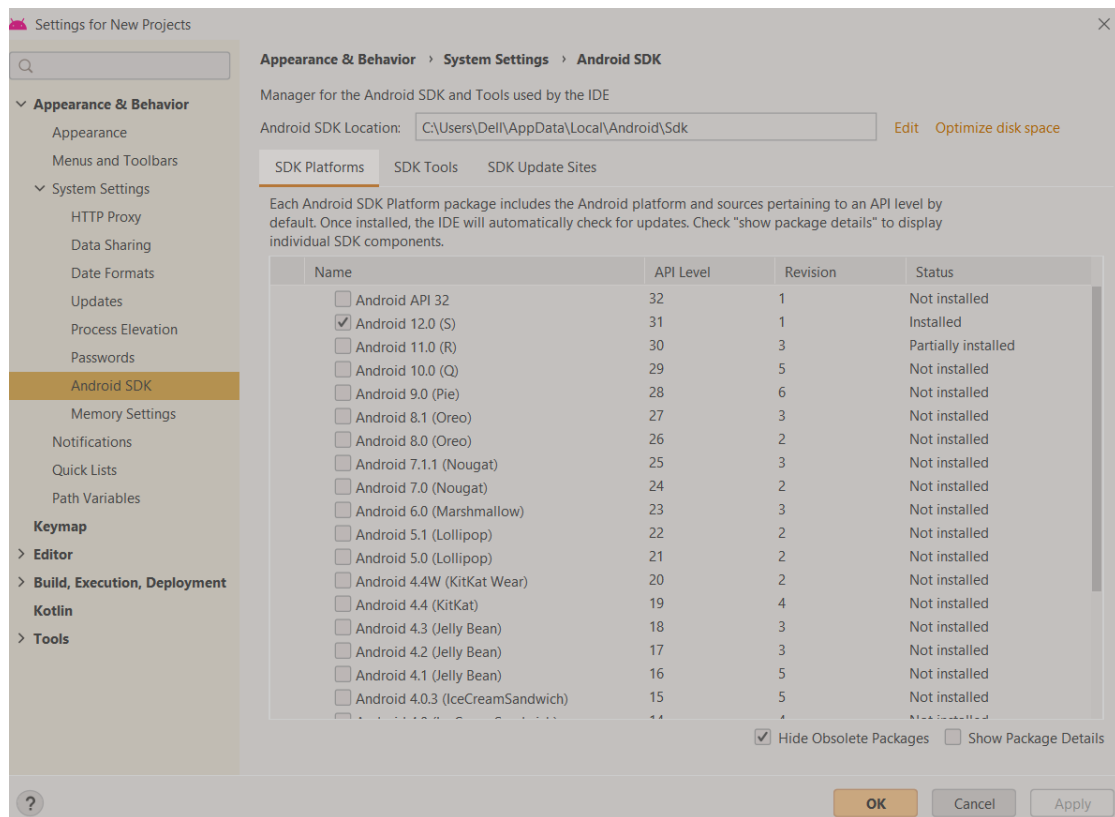
React Native Environment Setup



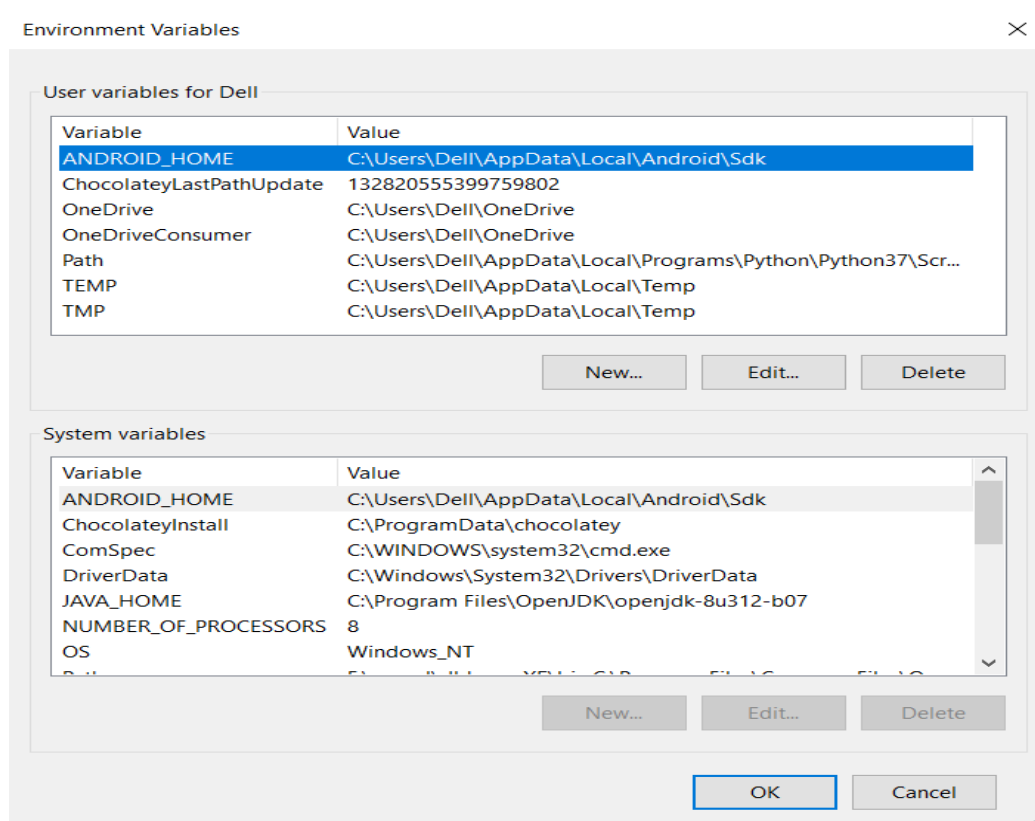
Installing Android Studio:

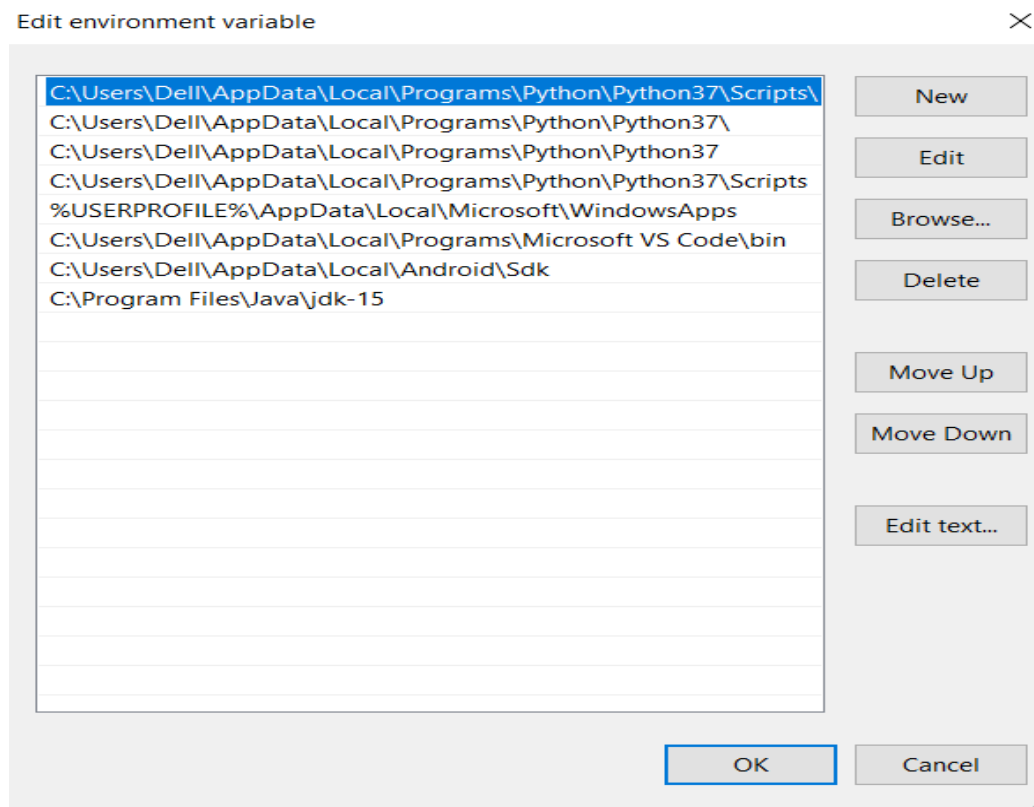


In More Actions you need to select SDK manager



After selecting this copy the Android SDK Location and set it in Environment Variables





Install React Native

Open Windows PowerShell where multiple Tabs can be used.

First, create new project

React Native has a built-in command line interface, which can be used to generate a new project.

`npm react-native init AwesomeProject(ProjectName)`

Started the project with a custom React Native template like TypeScript with `--template` argument:

`npm react-native init AwesomeTSProject --template react-native-template-typescript`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Shweta> npx react-native init AwesomeTSProject --template react-native-template-typescript|
```

```
#####
###      ###      ###      ###
##      ###      ###      ##
##      #####      ##
##      #####      ##
##      #####      ##
##      #####      ##
#####
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
###      ###      ###      ###
#####

Welcome to React Native!
Learn once, write anywhere

✓ Downloading template
✓ Copying template
✓ Processing template
✓ Installing dependencies

Run instructions for Android:
  • Have an Android emulator running (quickest way to get started), or a device connected.
  • cd "C:\Users\Dell\UshaVahiniApp" && npx react-native run-android

Run instructions for Windows:
  • See https://aka.ms/ReactNativeGuideWindows for the latest up-to-date instructions.

|
```


Step 2: Start your application

Let Metro bundler run in its own terminal, open a new terminal inside a React Native project folder.

And run **npm react-native run-android**

```
C:\WINDOWS\system32\cmd.exe x Windows PowerShell x + -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Shweta> cd ParkMe
PS C:\Users\Shweta\ParkMe> npm run android
info Running jetifier to migrate libraries to AndroidX. You can disable it using "--no-jetifier" flag.
Jetifier found 1254 file(s) to forward-jetify. Using 8 workers...
info JS server already running.
info Installing the app...

> Task :app:installDebug
Installing APK 'app-debug.apk' on 'V2037 - 11' for app:debug
Installed on 1 device.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.9/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 41s
121 actionable tasks: 2 executed, 119 up-to-date
info Connecting to the development server...
8081
info Starting the app on "96426678730038J"...
Starting: Intent { cmp=com.parkme/.MainActivity }
PS C:\Users\Shweta\ParkMe>
```

The application will be created in the mobile.




Steps to install node.js:

To install Node.js visit the link <https://nodejs.org/en/download/> and download node.js

Downloads

Latest LTS Version: 16.13.1 (includes npm 8.1.2)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer node-v16.13.1-x86.msi	 macOS Installer node-v16.13.1.pkg	 Source Code node-v16.13.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.13.1.tar.gz	

<https://nodejs.org/dist/v16.13.1/node-v16.13.1-linux-x64.tar.xz>

Download the Node.js of operating system as per the user requirement like mac, windows, Linux, Etc.

After installing node.js, to check the version of installed node.js use the below command in command prompt.

```
~ $node --version
```

Or,

```
~ $node -v
```

We get output as;

A terminal window titled 'pc-6@pc6: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'node -v' being executed, resulting in the output 'v16.13.1'. The prompt changes from '~ \$' to '~ \$' after the output.

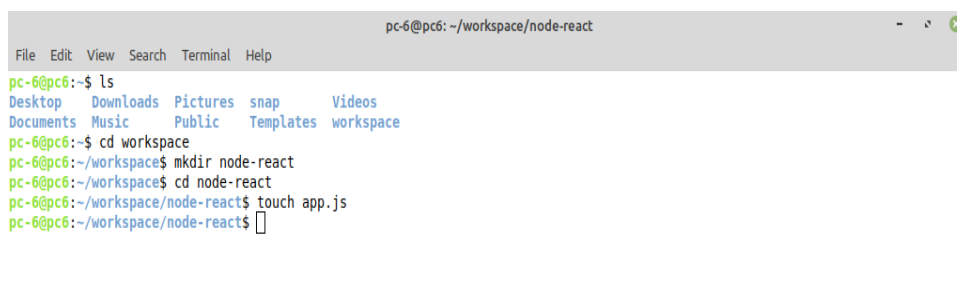
```
pc-6@pc6: ~  
File Edit View Search Terminal Help  
pc-6@pc6:~$ node -v  
v16.13.1  
pc-6@pc6:~$
```

Getting started to node.js:

We must create a file in which we can write the code and execute the program. Below are few commands to make directory, change directory and create file

```
~ $ls  
  
~ $cd folderName  
  
~/folderName $mkdir newfolderName  
  
~/folderName/newfolderName $touch filename
```

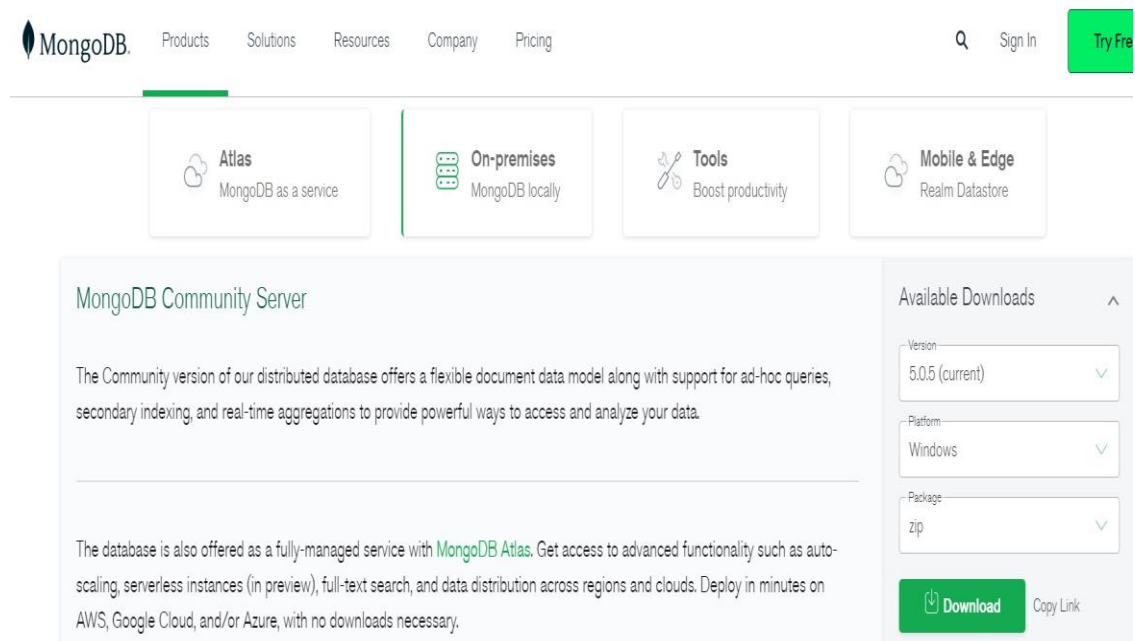
Now file is created as app.js in which we write our code and execute that file.

A terminal window titled 'pc-6@pc6: ~/workspace/node-react' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a series of commands: 'ls', 'cd workspace', 'mkdir node-react', 'cd node-react', and 'touch app.js'. The output shows the directory structure being created.

```
pc-6@pc6: ~/workspace/node-react  
File Edit View Search Terminal Help  
pc-6@pc6:~$ ls  
Desktop Downloads Pictures snap Videos  
Documents Music Public Templates workspace  
pc-6@pc6:~$ cd workspace  
pc-6@pc6:~/workspace$ mkdir node-react  
pc-6@pc6:~/workspace$ cd node-react  
pc-6@pc6:~/workspace/node-react$ touch app.js  
pc-6@pc6:~/workspace/node-react$
```

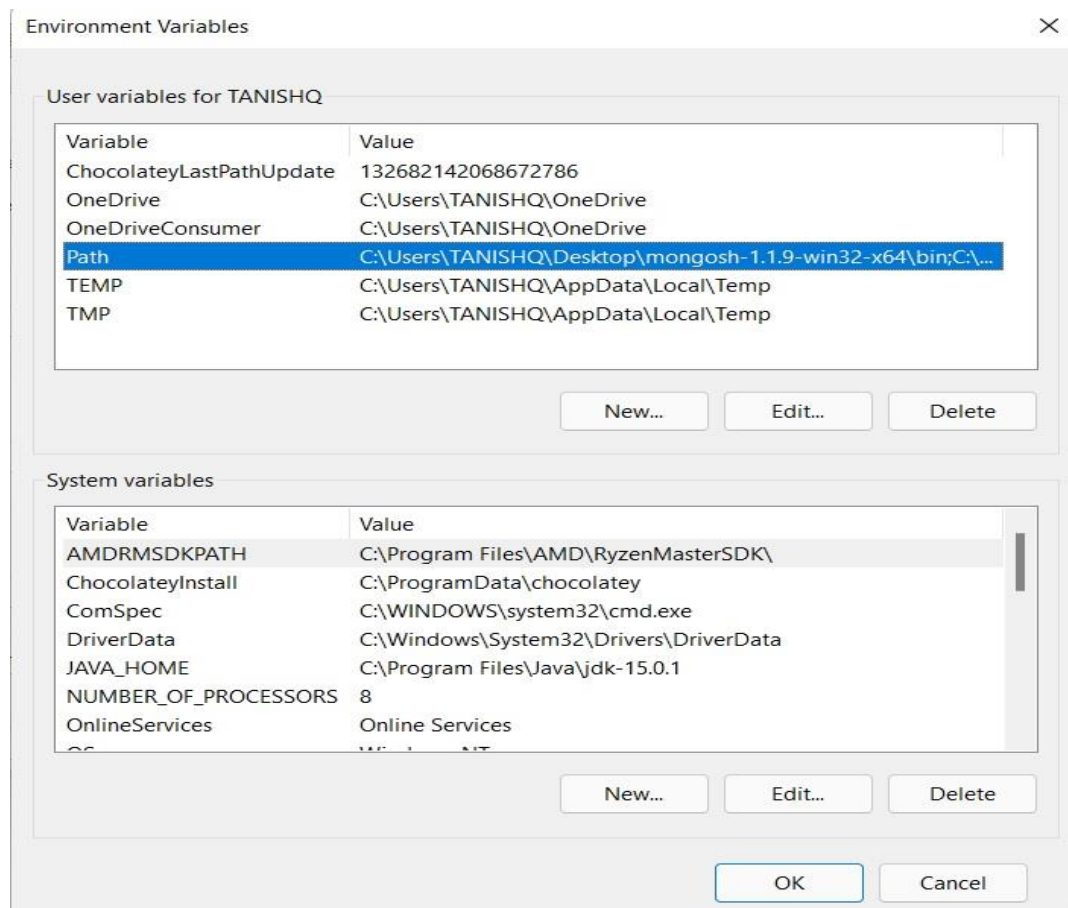
MongoDB Installation:

Download the file from MongoDB official site

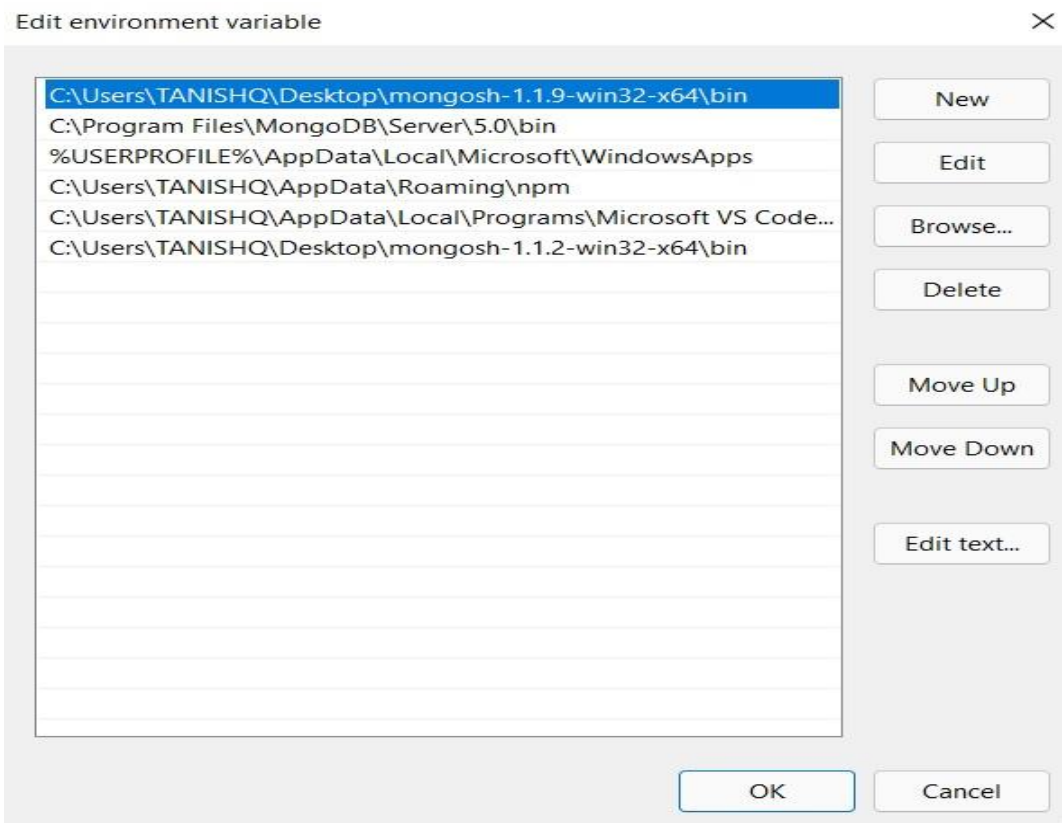


The screenshot shows the MongoDB official website. The navigation bar includes links for Products, Solutions, Resources, Company, and Pricing, along with a search icon, a Sign In button, and a Try Free button. Below the navigation bar, there are four main categories: Atlas (MongoDB as a service), On-premises (MongoDB locally), Tools (Boost productivity), and Mobile & Edge (Realm Datastore). The On-premises category is highlighted with a green border. Below these categories, the MongoDB Community Server is featured. It includes a description: "The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data." Below this, it states: "The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary." To the right, there is a section for Available Downloads. It includes dropdown menus for Version (5.0.5 (current)), Platform (Windows), and Package (zip). Below these dropdowns, there is a green Download button and a Copy Link button.

Set the permanent path for using mongoddb shell



The screenshot shows the Windows Environment Variables dialog box. The title bar is "Environment Variables". The dialog is divided into two sections: "User variables for TANISHQ" and "System variables". In the "User variables" section, the "Path" variable is selected and highlighted in blue. Its value is "C:\Users\tanishq\Desktop\mongosh-1.1.9-win32-x64\bin;C:\Users\tanishq\AppData\Local\Temp". Below the list of variables, there are buttons for "New...", "Edit...", and "Delete". In the "System variables" section, several variables are listed, including AMDRMSDKPATH, ChocolateyInstall, ComSpec, DriverData, JAVA_HOME, NUMBER_OF_PROCESSORS, and OnlineServices. Below this list, there are also buttons for "New...", "Edit...", and "Delete". At the bottom of the dialog, there are "OK" and "Cancel" buttons.



6. REFERENCE'S:

Web References

- a) <https://reactnative.dev/>
- b) <https://nativebase.io/>
- c) <https://reactnavigation.org/>
- d) <https://nodejs.org/en/>
- e) <https://www.postman.com/>
- f) <https://www.mongodb.com/>
- g) <https://www.mongodb.com/products/compass>
- h) https://www.mongodb.com/cloud/atlas/lp/try2?utm_source=google&utm_campaign=gs_apac_india_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624347&adgroup=115749713423&gclid=Cj0KCQiAubmPBhCyARIsAJWNpiMF5N6NScBIIi-oTOjqkm4mkGYmOmFx7Kv2IjQ426yGKV8wi6yrzEaAkfaEALw_wcB

IEEE paper references:

- a) Ashok, D., Tiwari, A., & Jirge, V. (2020). *Smart Parking System using IoT Technology*. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). doi:10.1109/ic-etite47903.2020.457
- b) Mohammadi, F., Nazri, G.-A., & Saif, M. (2019). *A Real-Time Cloud-Based Intelligent Car Parking System for Smart Cities*. 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP). doi:10.1109/icicsp48821.2019.8958
- c) Patil, B. K., Deshpande, A., Suryavanshi, S., Magdum, R., & Manjunath, B. (2018). *Smart Parking System for Cars*. 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE). doi:10.1109/icrieece44171.2018.90