CHAPTER 1
INTRODUCTION

1.1 Overview

The traffic rule is the most essential rule to be followed by every individual. Traffic rules are to be strictly followed for which various departments under the government are working. The traffic violators are fined based on their offense. There is different type of Indian penal code for different mode of traffic violations. There is a fine charged to the offender by the police. The charge may be paid instantly or by the court after a couple of days. The fine paid instantly is made with a paper receipt as an acknowledgment. This is the normal case for an offender. The police are subjected to maintain a large data of all offenders as a paper receipt which is a tedious process. The maintaining of each police at different stations is also a big deal for higher authority. The sections under the Indian penal code are to be made avail for a better secured nation. The police are working hard for the maintenance of proper traffic rules.

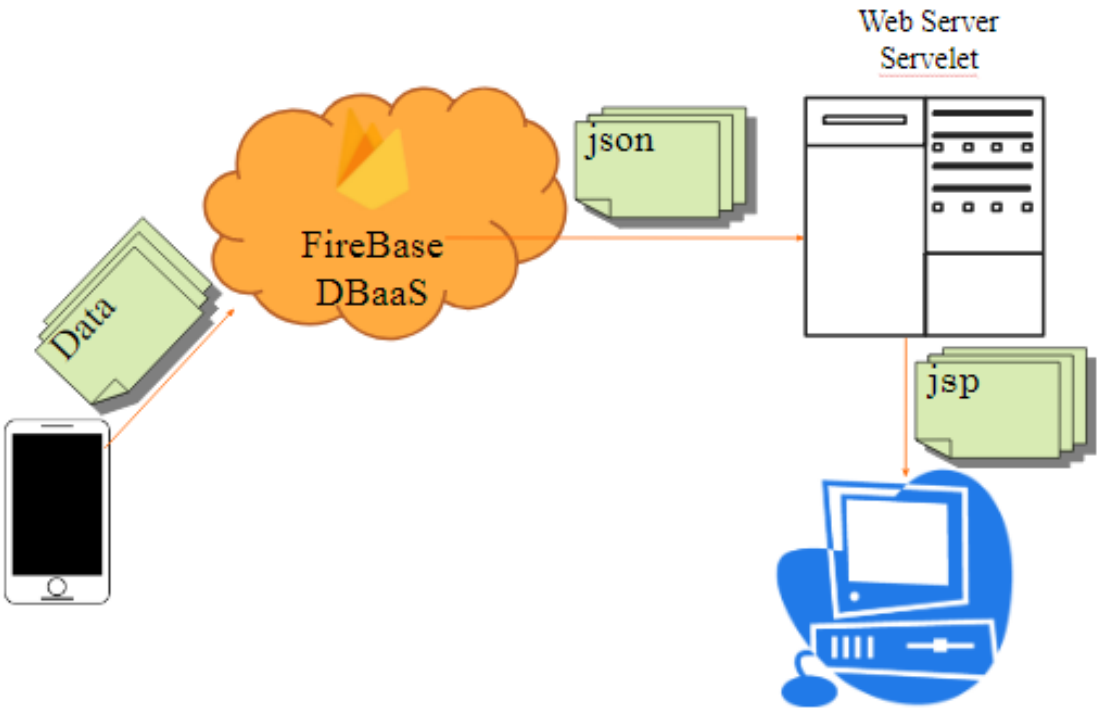


Figure 1.1 Architecture Diagram

1.2 Objective of the project

The number of traffic accidents has been increasing day by day in India. Due to this, the authorities in the traffic police department are forced to find ways that can reduce the number of accidents and make Indian roads safer. One of the major reasons for a high number of road accidents is that traffic rules are not being followed by the citizens of India. Driving on a road implies driving in a predictive manner where other drivers or pedestrians are aware of where you will be directing your vehicle. To make sure that people follow traffic rules even when nobody's watching, traffic police took the help of Electronic Traffic Challan System.

A challan is an official text written on a piece of paper issued to a person in particular. A challan is usually issued for violating traffic rules and regulation in India and neighboring countries. If a traffic challan is issued in your name, you are liable to pay the fine as mentioned in The Motor Vehicles Act, 1988. A traffic police officer holds the right to issue a challan in the name of any person who does not follow one or more traffic rules. An E-challan is nothing but an electronically generated challan with the help of Electronic Challan System.

E-challan has recently replaced the physical pieces of paper. From a person breaking a traffic rule in India to him/her paying the corresponding fine, several stages of this process have been digitized thanks to the introduction of the e-challan check. The traffic police have introduced the CCTV-enabled e-Challan system in an attempt to eventually curb the rising number of road accidents. A CCTV camera continuously records footage of the ongoing traffic. If a motorist breaks any traffic rule, the act will be recorded in the footage. The police will try to extract the number from the vehicle’s screen-shot captured from the CCTV footage and the offense will be

registered in the records. The traffic police will be then coordinate with the Regional Transport Office (RTO) to find the details of the vehicle and its owner. Details like the name and address of the vehicle owner make and model of the vehicle, etc. As soon as they receive the details, an SMS will be sent on the registered phone number of the violator. This SMS will contain the time, date and location of the offense. There are two ways in which you can pay an e-challan issued in your name. The most convenient way is to pay it online.

The authorities have created a dedicated website for paying the e-challan directly. You can easily find the e-challan payment online website for your Indian state. The e-challan issued in your name will have a unique challan number. Enter this number in the box provided on the website and proceed to make the payment. Another method is to pay the e-challan offline. You need to visit your nearest police station to make the payment against your e-challan number. Our project helps in dealing this existing system in a convenient way possible.

The project provides an extension for this existing system. The details about the offender are provided by a traffic inspector on spot and the information gets stored in a cloud database. This kind of storing of data enables faster retrieval and easy storage of data. Security is high in this system and it is easy to process. All it takes to do this a smartphone. After the completion of the offender rituals, the information will be entered in an app and gets submitted by the traffic inspector. This gets stored in a cloud database and the data are transferred to a website. The website processes the data and displays the results to the higher officials. Hence the data will be stored and monitored in a convenient manner.

1.3 Module Description

1.3.1 Application module:

This module is used by the traffic police as an android application. The traffic police is made avail with a separate username and password by which they can login to their account. The login is made in accordance with the centralized cloud server. The police man can use the application to store the offender details and record the offence. The details such as the license number, aadhar number or number plate can be noted for offender reference. After this reference the section number for the offence can be entered by the traffic police. The process takes till the final module where the fine is collected. The data entered here are updated simultaneously at the website module. The data are collected and stored by means of a cloud server.

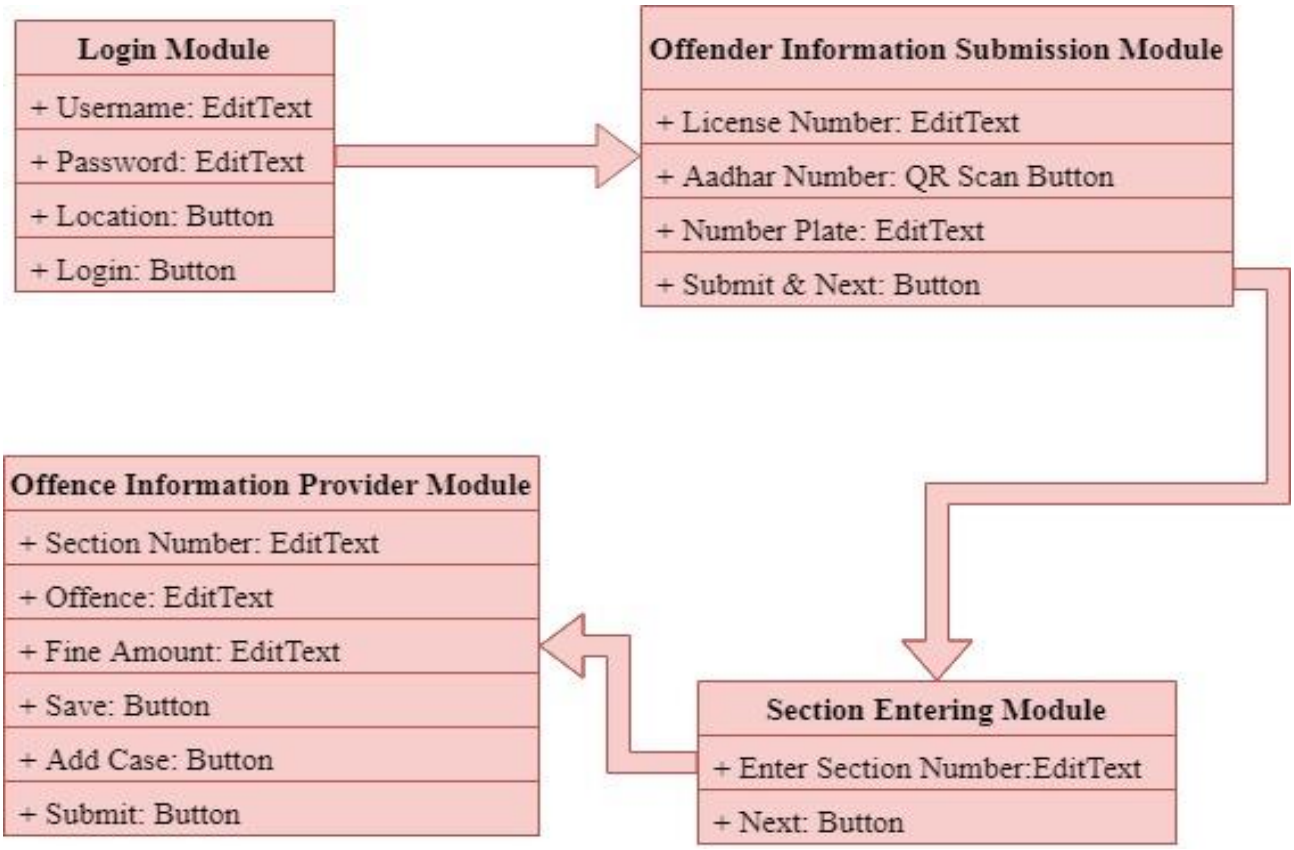


Figure 1.2 Overview of Mobile Application

1.4.2 Website module:

This module is based on the server side where all the data is stored. The system is made under cloud databases. In the cloud database environment, the cloud providers of server, storage and other infrastructures are responsible for the maintenance and availability of all the centralized data stored. In the DBaaS environment of cloud computing, the service providers are responsible for maintaining and operating database software, leaving the DBaaS users responsible only for their own respective data. This enables the centralized server maintenance more easy for the cops. This module is the essential module for the maintenance of cops in several stations. It also provides the

offender and offence details under different cops. In recent years, with the rapid development of video surveillance infrastructure, more and more intelligent surveillance systems have employed computer vision and pattern recognition techniques. In this paper, we present a novel intelligent surveillance system used for the management of road vehicles based on Intelligent Visual Internet of Things (IVIoT). The system has the ability to extract the vehicle visual tags on the urban roads; in other words, it can label any vehicle by means of computer vision and therefore can easily recognize vehicles with visual tags. The nodes designed in the system can be installed not only on the urban roads for providing basic information but also on the mobile sensing vehicles for providing mobility support and improving sensing coverage. Visual tags mentioned in this paper consist of license plate number, vehicle color, and vehicle type and have several additional properties, such as passing spot and passing moment. Moreover, we present a fast and efficient image haze removal method to deal with haze weather condition. The experiment results show that the designed road vehicle monitoring system achieves

an average real-time tracking accuracy of 85.80% under different conditions. Cloud Computing in the Automotive Domain Cloud computing has been planned to reshape transport package and services within the automative domain. As plenty of cars area unit equipped with devices which will access the net. propose to integrate existing transport networks, various sensors, on-board devices in vehicles, and cloud computing to create transport clouds.By victimization the standard approach to decompose a posh system into smaller subsystems consistent with their functions, we are able to divide a transport cloud service platform into variety of practical services and subsystems like traffic administration,service routing, IP, vehicle pledge analysis and mining, etc. As cloud computing includes 3 distinct services—platform as a service (PaaS), infrastructure as a service (IaaS) further because the well-liked package as a service (SaaS), a compound of SaaS, PaaS, and IaaS ought to be leveraged for building transport cloud service platforms. moreover, clouds may be divided into personal, public, and hybrid clouds. So transport cloud service platforms may be designed to be a hybrid cloud wherever some services, like user data question, are often hosted on public cloud platforms and alternative missing-critical services, like traffic

administration, ought to be hosted on personal cloud platforms. In this paper, we presented an interactive approach to digitalize the traffic fine system. The problem of normal receipt system is replaced by the digital system by means of the android application. Effective solution for acknowledgment from the traffic police is made through an SMS to the offenders. The main aim is to maintain a complete database for the maintenance of the cops and offence details at a large quantity. As the early system has challenges in maintenance of receipted data this proposed project may provide a better enhancement using with the cloud computing.

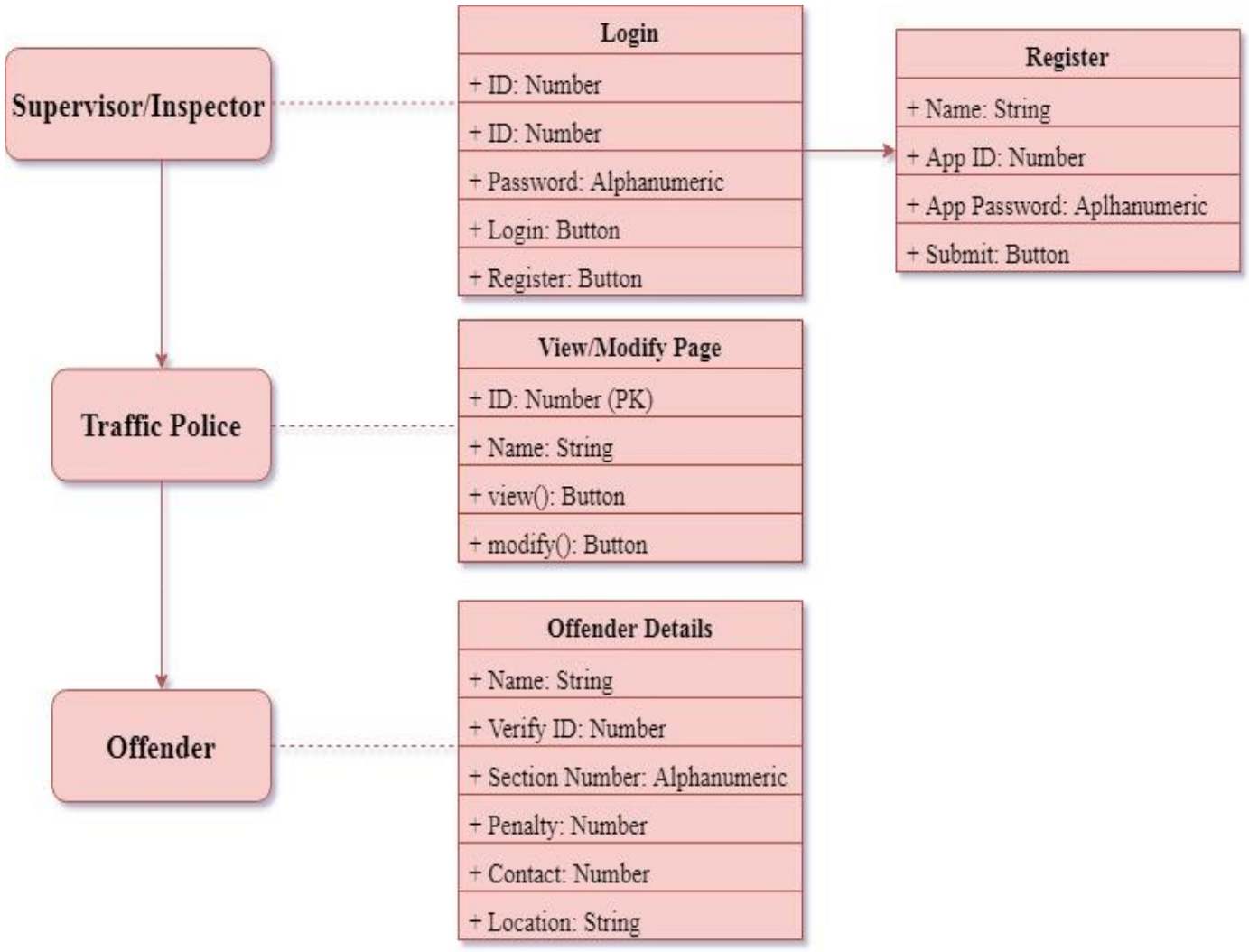


Figure 1.3 Overview of Website

CHAPTER 2

LITERATURE REVIEW

2.1 Study on Google Firebase for Website Development (The real time database):

A database is a collection of information that is organized in such a way which is easily manageable and accessible. Firebase is a “NoSQL” databases that are useful for large sets of distributed data. NoSQL databases are effective for big data performance issues that relational databases aren't built to solve. Along with this, firebase is also a “Real Time” database which provides an API that allows developers to store and sync data across multiple clients. Data in firebase is saved as json and can be exported. Implementing Firebase is quick and easy. With intuitive APIs packaged into a single SDK, we can focus on quickly building our app and not waste time building a complex infrastructure. With Firebase, you can focus your time and attention on developing the best app(s) possible for your business. Since the operations and internal functions are so solid, and taken care of by the Firebase interface, you can spend more time developing the high quality app that users are going to want to use more reliable way across platforms.

2.2 Application of Firebase in Android App Development-A Study

The server used for Android apps are Oracle SQL, Microsoft SQL Server, and MySQL which are connected to the server with PHP files. Then Firebase came into existence for Android apps which uses JSON for storing data. The other servers use a table (rows and columns) format for storing data Firebase is NoSQL based. There are very few cloud based server available which are similar to firebase, like: AWS

Mobile Hub- It is integrated console that helps to create, build, test, and monitor the mobile apps that leverages AWS services. CloudKit- It is an Apple framework which helps to save data and store assets but similar to iOS only. Parse Server- It was released by Facebook to replicate functionality of Parse which is an open source server. This is no longer in existence as Facebook shutdown this project. Firebase is considered as web application platform. It helps developers‟ builds high-quality apps. It stores the data in JavaScript Object Notation (JSON) format which doesn't use query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data.

2.3 Retrieve Real-time Data in Android Using Firebase

Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Obtaining data from a database management system: In this case, it is considered that data is represented in a structured way, and there is no ambiguity in data. In order to retrieve the desired data, the user presents a set of criteria using a query. Data retrieval is the process of identifying and extracting data from a database, based on a query provided by the user or application. It enables the fetching of data from a database in order to display it on a monitor and/or use within an application.

2.4 Making It Work for Everyone: HTML5 and CSS Level 3 for Responsive, Accessible Design on Website

This article argues that accessibility and universality are essential to good Web design. A brief review of library science literature sets the issue of Web accessibility in context. The bulk of the article explains the design philosophies of progressive enhancement and responsive Web design, and summarizes recent updates to, HTML5and CSS Level 3. The final section of the article walks readers through the Web site creation process. The tools and techniques described in the article can be used to create a library Web site which can be accessed equally by all patrons without sacrificing aesthetics or usability at any level.

2.5 A Comprehensive analysis of XML and JSON web technologies

In this global era, internet plays a vital role to share information all over the world. There are some standard protocols and web technologies to represent the information on internet like HTML, JavaScript, ASP, JSP, XML, JSON etc. All these standards have pros and cons, and also depend on the requirement that which exchange of format more reliable. This research work shows the comparative analysis of XML and JSON using Multi-criteria Decision Support System MCDS, and analyzes use of web technology according to the requirement need. Today web technology is on the level that we are talking about synchronous and asynchronous transformation technologies like XML, AJAX, .NET, JSON and so on are in practice. But 15-16 years ago the people was only thinking how to display documents on the internet and for that purpose HTML (Hyper Text Markup Language) had been developed. And then with the time as people get use to with the internet, then the user start demands for new things like to have interactive websites that they can display information as well as can take input from user for that purpose different technologies had been

developed that we are using in dynamic websites such that ASP, JSP, PHP, ASP.NET etc. then there was a problem how to transfer data between different platforms so XML has been developed which store plain text base data independent of platform and can easily transfer data between cross platforms.

2.6 Android Based Mobile Application Development And Its Security

In the advancing world of technology, Mobile applications are a rapidly growing segment of the global mobile market. Mobile applications are evolving at a meteor pace to give users a rich and fast user experience. In this paper, Android mobile platform for the mobile application development, layered approach and the details of security information for Android is discussed Google released Android which is an open-source mobile phone operating system with Linux-based platform. It consists of the operating system, middleware, and user interface and application software. Certainly, Android is about to become the most widely used OS on mobile phones, but with Android comes a security

vulnerability that few users take into account. On Android Market, where you can download thousands of applications for Android, anyone can upload their programs without having to submit them to careful security checks. This makes Android a prime target for computer criminals. In this paper, we discuss a layered approach for android application development where we can develop application which downloads data from the server. Also an Android Application Sandbox (AASandbox) which is able to perform both static and dynamic analysis on Android programs to automatically detect suspicious applications is also discussed.

2.7 Database in Cloud Computing - Database-as-a Service (DBaas) with its Challenges

The rise of cloud computing has had an impact on multiple industry sectors – most notably server and storage infrastructure – as enterprises explore and enjoy the potential cost and agility benefits that come with using virtual infrastructure that is available on demand and as required. The data processing and analytics sectors are no exception, and 451 Research has observed the emergence of growing number of database-as-a-service offerings, as well as growing adoption by startups and established enterprises alike. Database-as-a-service offers multiple potential benefits, including lower database licensing and infrastructure costs, faster time to application development, and reduced administration overheads. These benefits are most likely to be experienced by database administrators and architects, although senior decision-makers and business users also stand to gain from having on-demand access to database services, rather than waiting for databases to be configured and deployed on dedicated physical or virtual server infrastructure. While 451 Research anticipates growing adoption of database-as-a-service (DBaaS), adoption is currently nascent compared with other cloud services, as enterprises look to make the most of their investments in on-premises database deployments, and also to identify the most appropriate workloads for transition or migration to DBaaS. This report explores the factors shaping those adoption trends, including the potential benefits and challenges to DBaaS adoption, the economics of the cloud as they relate to database workloads, and adoption lifecycle.

2.8 Cloud Database Database As A Service

Cloud computing has been the most adoptable technology in the recent times, and the database has also moved to cloud computing now, so we will look into the details

of database as a service and its functioning.This paper includes all the basic information about the database as a service. The working of database as a service and the challenges it is facing are discussed with an appropriate. The structure of database in cloud computing and its working in collaboration with nodes is observed under database as a service. This paper also

will highlight the important things to note down before adopting a database as a service provides that is best amongst the other. The advantages and disadvantages of database as a service will let you to decide either to use database as a service or not. Database as a service has already been adopted by many e-commerce companies and those companies are getting benefits from this service

2.9 QR Code Scanning app for Mobile Devices

QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix bar code first designed for the automotive industry . Today the QR code is widely used in all industries. In our paper we present an implementation of an Android device using libraries is and combined algorithms in order to be able to scan any QR code fast accurate and easy. The devices that we targeted for our application are the Google Glasses and an Android operated phone. The implementation for each of the devices was slight different, but the core algorithms and libraries were the same. A barcode is a machine-readable optical label that contains information about the item to which it is attached. The bar code can best be described as an "optical Morse code". Series of black bars and white spaces of varying widths are printed on labels to uniquely identify items . The bar code labels are read with a scanner, which measures reflected light and interprets the code into numbers and letters that are passed on to a computer. In order to have a bar code that was small in size, easy to read and capable of holding both a large amount of data and a large variety of character types, the market called for a new approach. Enter 2-Dimensional bar

codes. The 2-Dimensional barcodes offers good advantages like high density, high capacity, error detection, small areas. PDF417 is a stack of two-dimensional barcode. The matrix barcode is that codes the data based on the position of black spots within a matrix. Each black element is the same dimension and it is the position of the element that codes the data.

2.10 A Comparative Study between Dynamic Web Scripting Languages

With security, performance, manageability, and scalability requirements for enterprise applications rapidly increasing, choosing the best scripting language in today's competitive environment remains imperative. Scripting language are improving on every front, but enterprises continue to face the arduous but necessary task of evaluating solutions to find the best fit for their unique needs. The paper tends to analyse and compare the performance of the two most popular dynamic scripting languages used in web development, the PHP and ASP.NET. The comparison is based on testing and comparing the performance of the two technologies using software called WAPT. We measure the performance of these two languages using one application benchmarks; an online bookstore. This similarity helps to establish realistic comparison of applications performance. WAPT provides a load and stress testing a tool that provides an easy to use, consistent and cost effective way of testing websites and intranet applications with web interface.

2.11 A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software

A web application is a distributed application that runs on more than one computer and communicates through a network or a server. Specifically, a web application is accessed with a web browser as a client and provides the ability to update and

maintain a program without deploying and installing software on client computers. Web programmers today are confronted with the difficulty of having to work with constantly changing technologies, and having to make the right choice of which development technology to use. This paper presents a comparison of web application development technologies using open source software and proprietary software. The comparison involves three major web development technologies namely: Java Server Pages (JSP), Active Server Pages (ASP.NET) and PHP Hypertext Preprocessor (PHP). For the comparison to take place, a web application was developed in all three technologies using the same requirements. The model-view-controller (MVC) design pattern was employed in developing the web application, and the parameters for the comparison are cost of implementation, browser compatibility, response time of HTTP requests, operating system compatibility and mobile platform compatibility.

2.12 Mobile Application Development: All the Steps and Guidelines for Successful Creation of Mobile App: Case Study

Gone are the days when the mobile phone had to ring to capture our attention or the computer was the only device people used. The mobile application field has been rising at a tremendous rate with the drastic increase in the number of mobile apps in various mobile phones and tablets. Mobile apps are essential as they provide functionalities that can server useful purposes such as finding a location or booking movie tickets online. In today's fast paced world, mobile marketing is becoming very competitive. To ensure visibility of your app in such complex scenario, a specific approach needs to be followed to ensure a successful app development. In this paper, numerous factors that can play a significant role in successful app development are discussed with specific examples and explanation. Mobile

Application Development refers to the process of making application software for handheld devices such as mobile phones and Personal Digital Assistants. Through the usage of mobile apps, the user is provided with various features that will enable him to fulfill all his needs and much more. Apps should be interactive to the users. Apps can be downloaded from various platforms such as Google Play Store and iOS App Store. There are free apps as well as paid apps. Some apps can be used for free for a specific amount of time before subscribing for premium membership. For apps with a price, about 20%-30% goes to the distribution provider(Example-iTunes) and the rest to the production.

2.1 Literature Survey Table

No.	Name	Description	Pros	Cons
1	Study on Google Firebase for Website Development. (The real time database) (Hari Shankar Sing and Uma Shankar Singh)	Firebase is a "NoSQL" database which are useful for large sets of distributed data.	•Scalable. •Hierarchical Storage .	•Internet Connection is needed.
2	Application of Firebase in Android App Development -A Study. (Chunnu Khawas and Pritam Shah)	The server used for Android apps are Oracle SQL, Microsoft SQL Server, and MySQL which are connected to the server with PHP files.	Has real-time and cloud-based database where you can store data.	Querying and aggregating are limited compared to SQL
3	Retrieve Real-time Data in Android Using Firebase. (Shravan I.V.)	Firebase is a cloud based platform for mobile and Web application development.	Realtime/streaming updates are pretty easy.	Firebase is somewhat deprecated in favor of Cloud Firestore.
4	Making It Work for Everyone: HTML5 and CSS Level 3 for Responsive, Accessible Design on Website. (Stewart C. Baker)	Explains the design philosophies of progressive enhancement and responsive web design, and summarizes recent updates HTML5 and CSS Level 3.	Allows storing files or web application makes them accessible without internet connection.	Considerable effort even if wishes to carry out a minor change in the site.

No.	Name	Description	Pros	Cons
5	A Comprehensive analysis of XML and JSON web technologies. (Zia Ul Haq, Gul Faraz Khan and Tazar Hussain)	XML and JSON is compared on the basis of these comparisons that which technology is required for whom and for what purpose.	Concise format thanks to name/value pair - based approach.	No namespace support, hence poor extensibility.
6	Android Based Mobile Application Development And Its Security. (Suhas Holla and Mahima M Katti)	Android mobile platform for the mobile application development, layered approach and the details of security information for Android is discussed.	Budget-friendly option.	Endangering data privacy.
7	Database in Cloud Computing - Database-as-a Service (DBaas) with its Challenges. (Shweta Dinesh Bijwe and Prof. P. L. Ramteke)	A DBaaS promises to move much of the operational burden of provisioning, and access control from the database users to the service operator, offering lower overall costs to users.	The database is off site, meaning loss of power or natural disaster at your business doesn't affect it.	Don't have direct access to the servers that are running your database.

No.	Name	Description	Pros	Cons
8	QR Code Scanning app for Mobile Devices. (Mircea Moisoiu, Andrei Negrau, Robert Gyorodi, Cornelia Gyorodi and George Pecherle)	Present an implementation of an Android device using libraries and combined algorithms in order to be able to scan any QR code fast accurate and easy.	Ease of use. Range of uses. QR codes are traceable.	Dependability on a mobile device or Smartphone. Lack of awareness.
9	A Comparative Study between Dynamic Web Scripting Languages. (Alok Ranjan, Rajeev Kumar and Joydip Dhar)	Compared the impacts of these three languages on the performance of a web server.	Dynamic content generated for each user.	Expensive -setting up, maintaining, and scaling a database is costly.
10	A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software. (David A. Botwe and Joseph G. Davis)	Presents a comparison of web application development technologies using open source software and proprietary software.	The source code is not shared with the public for anyone to look at or change.	Many people have access to the source code of open source software, but not all of them have good intentions.
11	Mobile Application Development: All the Steps and Guidelines for Successful Creation of Mobile App: Case Study. (Kishore Baktha)	Numerous factors that can play a significant role in successful app development are discussed with specific examples and explanation.	Compatibility. Support and Maintenance.	Offline. Access Convenience.

Summary

The above table represents the Title, Description, Pros and Cons. That is used for easy understanding of the project and the methodology which can produce a clear idea about the project. Then the merits can create an impact on real-time implementation. The major objective of the project and disadvantages are used to understand the limitations of the project.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 Hardware Specification

- Monitor
- CPU with 4GB RAM
- Processor with 1.8 GHZ
- Smartphone with Android OS

3.2 Software Specification

- Android Studio v3.3.0
- Emulator – AVD
- Eclipse Oxygen v4.7
- Sublime Text Editor v3.0

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 Existing System

In today's world, there are models for recording the speed of vehicle, amount of alcohol consumed by a driver of a vehicle, capturing vehicle plate number using CCTV Camera etc but there is never a model to monitor and record the data of the offender that is the traffic violator's information. Also this storing and accessing of the data should be faster and efficient. Now-a-days, the simplest way to get the record is by Visiting your local county courthouse or DMV in person. This will probably get you where you need to go; if not, a look online at the county website, or a phone call to the DMV information line in your area, will be helpful as well. There is sometimes a small processing fee for requested records, and in many cases there is a waiting period, depending on where the records are stored and how long it takes the clerks to access them for you. In most cases, if the record is not available to you in person on the same day, it will be mailed to you within the few weeks following the request. This is a long period and people never want to wait for that long. Hence the notification should be sent to the offender as soon as possible via email address or SMS. This is a more complex way of getting your traffic violation records, but if for some reason requesting them from the DMV is not possible or not desired, then technically you can achieve the same result doing it this way. Traffic violations are considered public record, meaning the information is available to anyone who wants it, and is typically published in some sort of database. Many facts of public record are published in local newspapers, such as police reports or Public Notice columns, where printing the information constitutes notification of the public. So, if you have some time on your hands and access to some sort of database of public records and notices in your area, you can probably find your traffic violations this way. There is a

summarization called a Driver Abstract that is also public record, which contains a list of violations and other public information for each licensed driver in the area. If you can find a database of Driver Abstracts and locate yourself within it, it'll be much simpler to find what you need. These are the procedures to be followed to get

access for knowing your violation information. When these are emailed to you at the exact time when you commit any offense, this hectic process of getting the details will be avoided.

4.1.1 Disadvantages in the existing System

- Using the traditional database for storing the offender's offense information will make the storing and retrieval of data more difficult.
- The existing system does not have a better authentication and verification mechanisms such as QR scan.
- When the offender wants to access his/her traffic violation details, he/she doesn't have to follow a hectic process to retrieve it.
- There is always an issue of occurrence of error because it doesn't use any latest mechanisms.
- The traffic monitoring should have a data recording system as its extension

4.2 Proposed System

The normalized manual method of fine collection and maintenance of traffic police is digitalized in this proposed system. The method by which paper receipt used is been replaced by digitalized method. The replacement is done by an android application and a website. The android application is used by the police for fine collection from

the offender. There is a web site at with updating is made dynamic. The offender information and the fine collection are updated dynamically. The web site is a centralized system where the information on android application at various stations is updated. The proposed system has the capability to send acknowledgement via SMS system for every offender. The SMS system is proposed by means to cover all the class of people. The acknowledgement provides a better way to know about the fine at which they pay. The centralized system is also updated simultaneously. The updated data provide the understanding about the number of offenders and the amount of fine collected per day.

4.2.1 Advantages of Proposed System

- Acknowledgement is received to every offender.
- The rate of offence rate can be calculated more easier.
- Maintaining the centralized cloud database is achieved.
- Monitoring of traffic police at station level is made avail.

4.2.2 Block Diagram

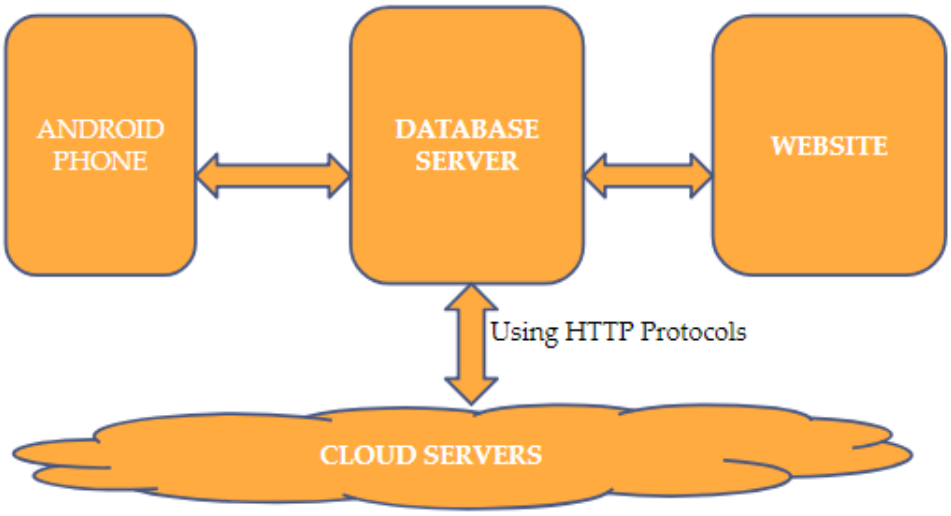


Figure 4.1 Block Diagram

MODULE DESCRIPTION

5.1 Types of Modules

5.1.1 Mobile Application Modules

Login Module:

The Traffic Inspector is prompted to enter his/her App ID and App Password provided by the Supervisor in the website registration.

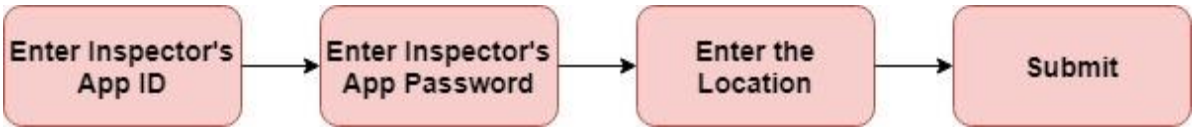


Figure 5.1: Login Module - Block Diagram

Offender Information Submission Module:

The Traffic rules violator's information gets checked and stored in this module. QR Scanning is used here for checking the aadhar details of the offender.

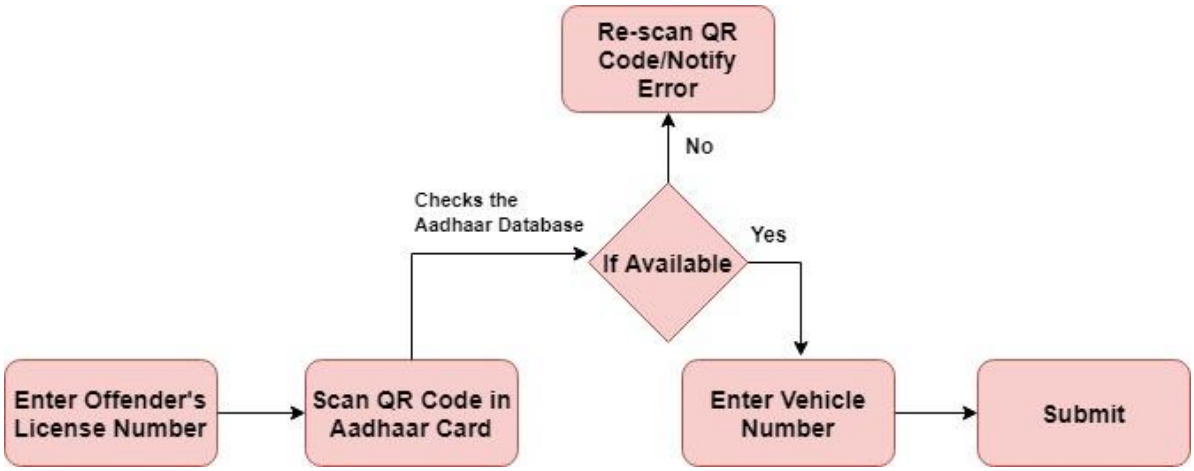
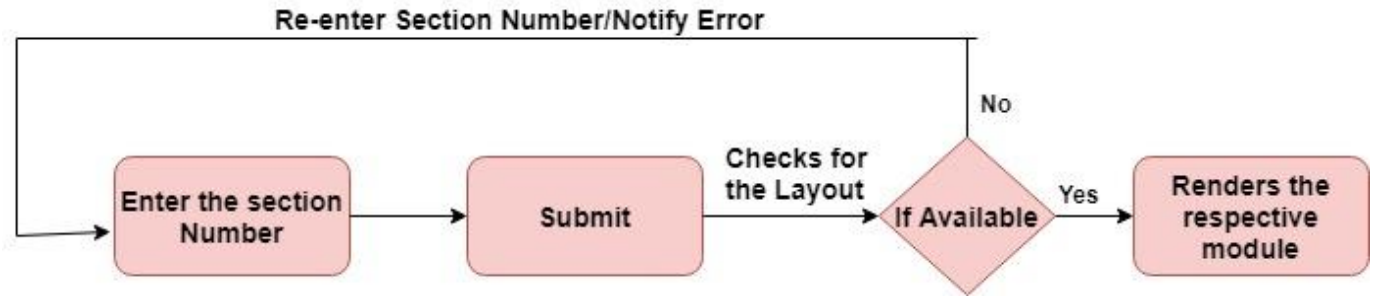


Figure 5.2 : Offender Information Submission Module - Block Diagram

Section Entering Module:

Every Traffic Violation has a specific Section Number. By providing the offender related section number, it will guide the traffic inspector to the corresponding Offence information submission module.

Fig 5.3: Section Entering Module - Block Diagram



Offence Information Submission Module:

The information of the Offence gets recorded and stored here in this module.

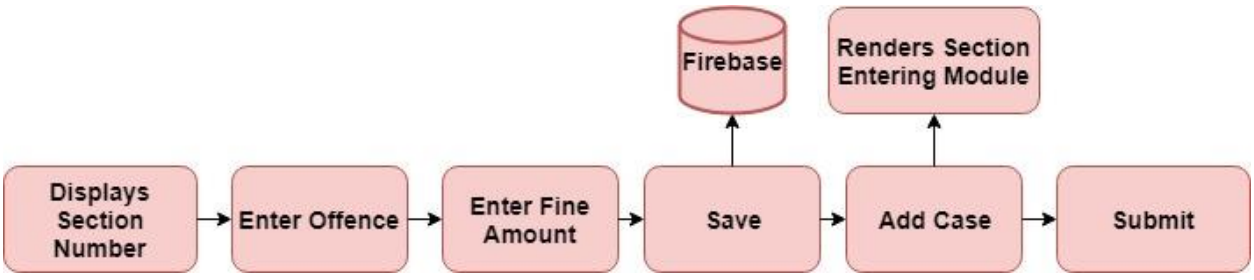


Figure 5.4: Offence Information Submission Module - Block Diagram

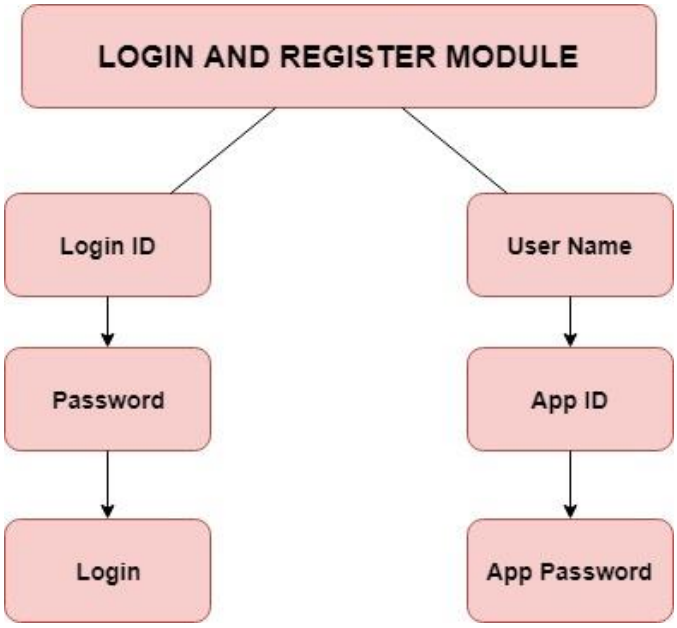
5.1.2 Website Modules

Login and Register Module:

The Supervisor has the authority over this module (Web Page).

He /She can add inspectors and give them their specific App ID and App Password.

Fig 5.5: Login and Register Module - Block Diagram



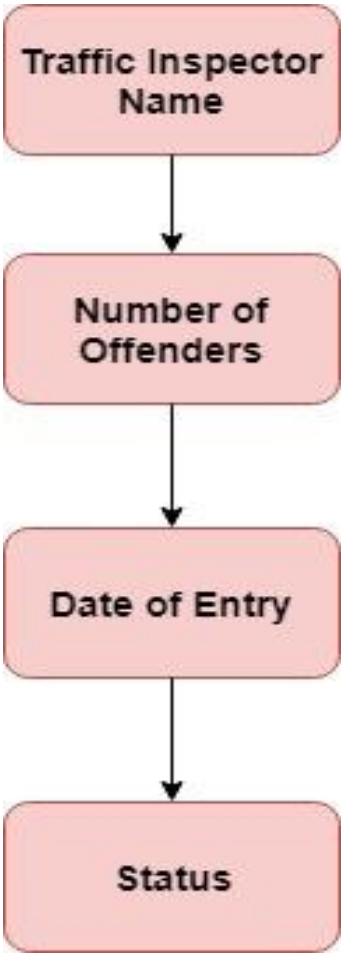
View/Modify Module:

Displays the contents of each inspector.

The supervisor can click on any inspector to view the details of the offender.

The Inspector's details stored in MySql Database is are retrieved by a PHP script.

Fig 5.6: View/Modify Module - Block Diagram

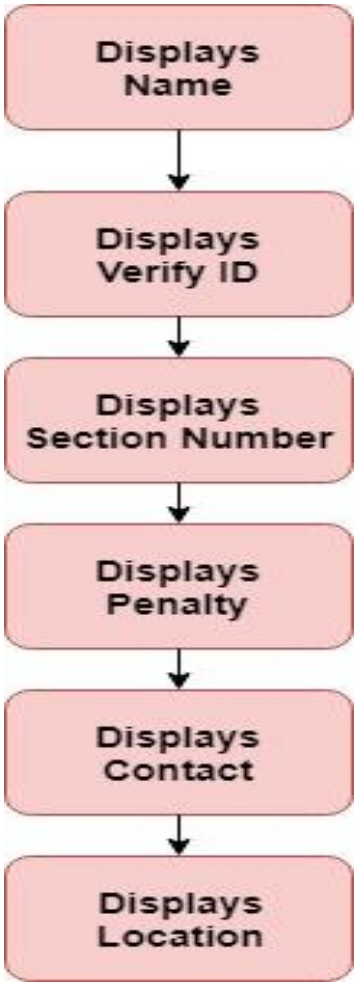


Offender Information Module:

Offender's informations are displayed here in this module.

It retrieves the JSON data from the Firebase Database and displays the HTML content in this dynamic web page.

Fig 5.7: Offender Information Module - Block Diagram



5.2 Firebase – Realtime Database Structure

5.2.1 Firebase

Firebase is a NoSQL cloud database. A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means *other* than the tabular relations used in relational databases. The Firebase Realtime Database is a cloud-hosted database. Data is stored as **JSON** and synchronized in realtime to every connected client.

Key Features of Firebase Database

Real-time

Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

Offline

Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

Accessible from Client Devices

The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

Scale across multiple databases

With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

Real World Usage

I've personally used Firebase in number of applications and it is very fluid in its working. It is highly efficient if you want a real-time database that is synced across every client.

5.2.2 Setting up a project

Go to the Firebase website and sign up for an account. Log in with a Google account for easy access. Go to the Firebase console. Go ahead and create a new one by clicking the blue **CREATE NEW PROJECT** button. Give new project a name. Check the URL at the top of browser **https://console.firebase.google.com/project/teamten/overview**, where the part of the URL after **/project/** matches the project's name. Project console window can be accessed from the following URL. **https://console.firebase.google.com/project/your-project-id/database/data**.

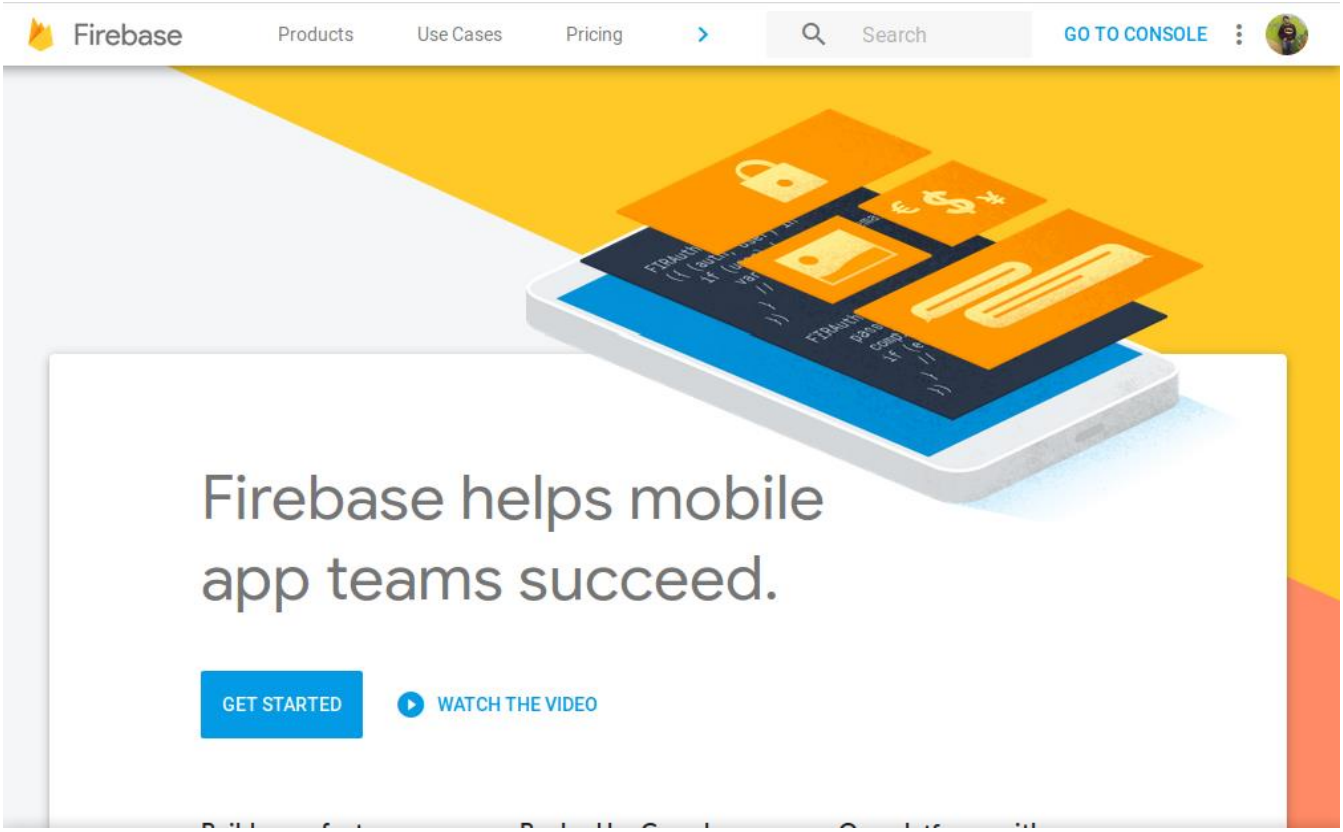


Figure 5.8: FireBase Front Screen

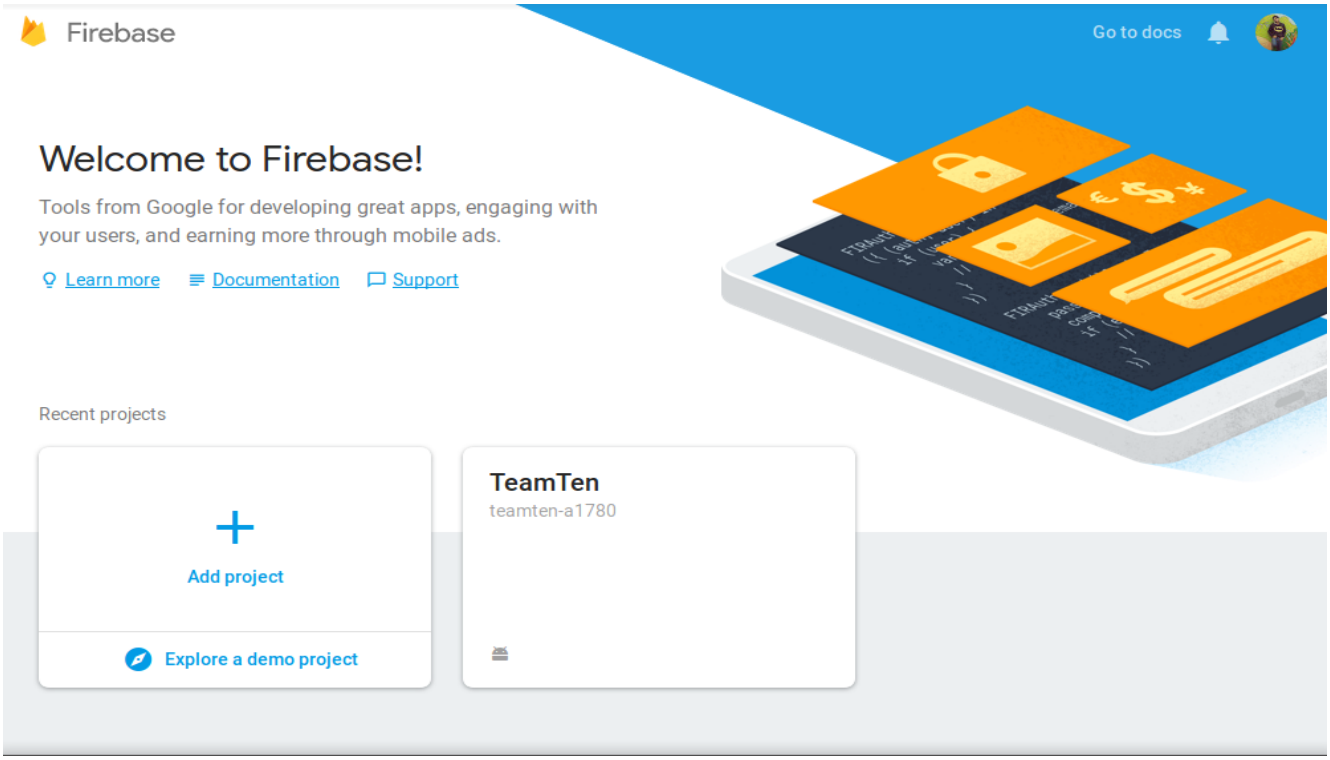


Figure 5.9: FireBase Project Explorer

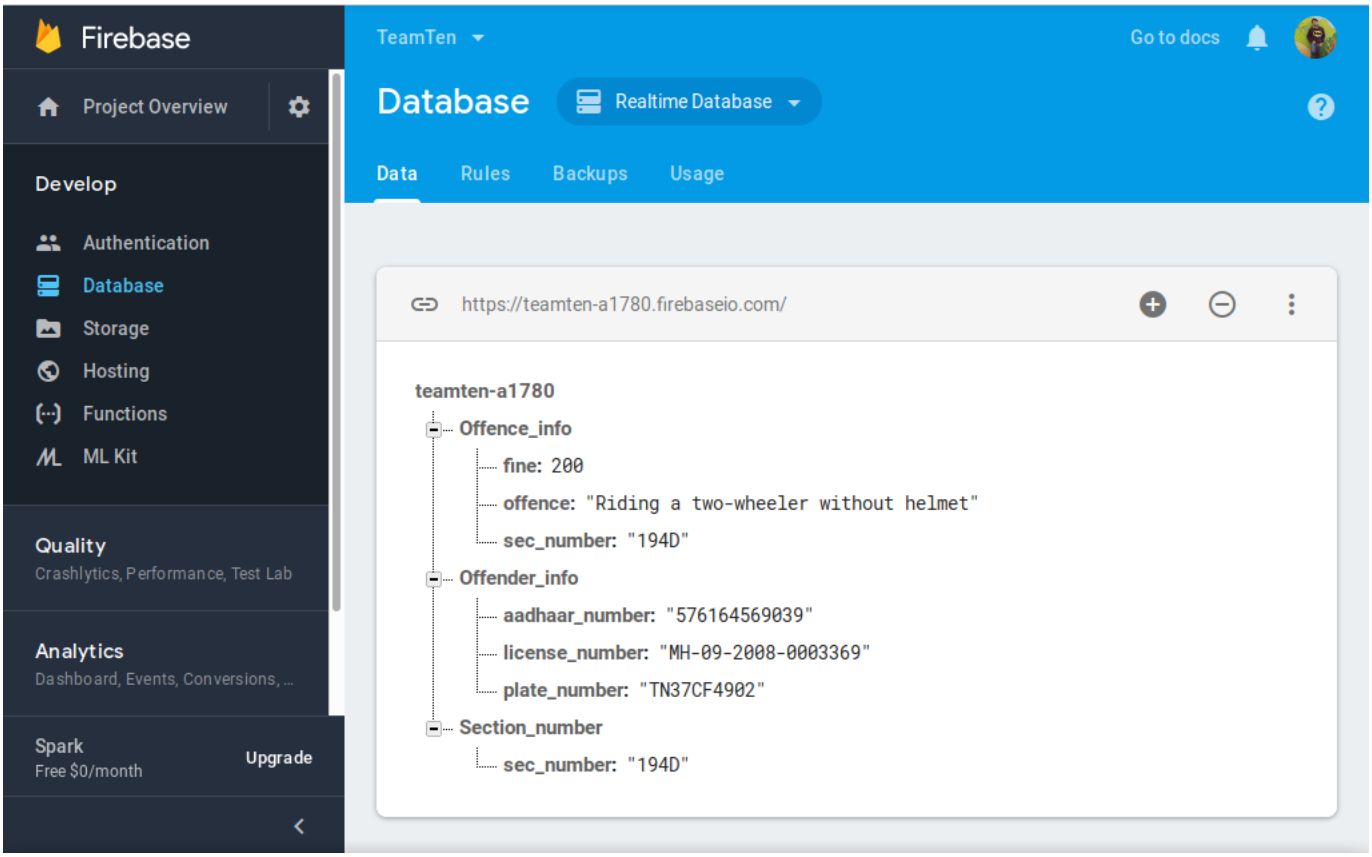


Figure 5.10: FireBase RealTime Database

5.2.3 Connecting the Database to the Mobile Application

It's time to add Firebase to the Mobile app.

To create a Firebase project:

1. Go to the Firebase console.

2. Click **Add project**, then select or enter a **Project name**.

Follow the remaining setup steps in the Firebase console, then click Create project

Firebase automatically provisions resources for the Firebase project. The process typically takes a few minutes. When the process completes, the overview page for the Firebase project in the Firebase console will be displayed.

Add Android app to it:

1. Click **Add Firebase to the Android app** and follow the setup steps.
2. When prompted, enter the app's package name. It's important to enter the package name the app is using.
3. Add the Firebase Android configuration file to the app:
    1. Click **Download google-services.json** to obtain the Firebase Android config file (google-services.json).
    2. Move the config file into the same directory as the root-level build.gradle file.

5.3 MySQL and PHP – Database for Website

5.3.1 XAMPP Server

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends,consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

-> The Apache HTTP Server, colloquially called Apache, is the world's most used web server software.

-> MariaDB database-*MariaDB* is a community-developed fork of the MySQL relational *database* management system intended to remain free under the GNU GPL.It's made by the original developers of MySQL and guaranteed to stay open source. Notable users include Wikipedia, WordPress.com: Create a website or blog and Google.

-> There are interpreters for scripts written in the PHP and Perl programming languages.

It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well. It's an Open Source web server with all the tools and language support built-in to it. It's ready to use and makes it easier for developers to take their code and host it locally and test the same.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.

XAMPP has the ability to serve web pages on the World Wide Web.

A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

5.3.2 Installing MySQL and PHPMyAdmin

We look into step by step process to install XAMPP for Windows. For Other Operating Systems, installation steps are similar.

- Download the XAMPP installer at http://www.apachefriends.org/en/xampp-windows.html
- Installation XAMPP is just like installing any other windows program. There are however, a few things that we must note.
- After you have downloaded XAMPP, run the setup. The warning message dialog window shown below appears.

htdocs- this is the web root directory. All of our PHP codes will be placed in this directory.

Mysql -this directory contains all the information related to MySQL database engine, by default it runs on port 3306.

php - this directory contains PHP installation files. It contains an important file named php.ini. This directory is used to configure how PHP behaves on your server.

By default, the Apache web server runs on **port 80**. If port 80 is taken by another web server, you can use a different port number.

XAMPP Control Panel

1) This section lists the installed services, modules and the process IDs PID(s). A green tick means the module has been installed as a service. The red mark means it has not been installed as a service. To install a service, click on the red mark. If the button shows a green tick and you click on it, the control panel will ask you if you want to uninstall the system.

2) This section shows Port(s) associated with the modules. The actions section is for;

1. Starting and stopping modules
2. Open the administrative windows for Apache and MySQL
3. Open configuration files for Apache, MySQL etc. to make changes
4. View log files for the modules

3) This section contains useful utilities such as Netsat, windows services short cuts etc.

4) This section displays status information on the modules. The control panel can be used to;

- Install and uninstall services such as Apache, MySQL etc. that are installed via XAMPP
- Start and stop services.
- Open configure files etc.

5) PHPMyAdmin comes prepacked with MySQL installation. If not, it can be downloaded from its official website.

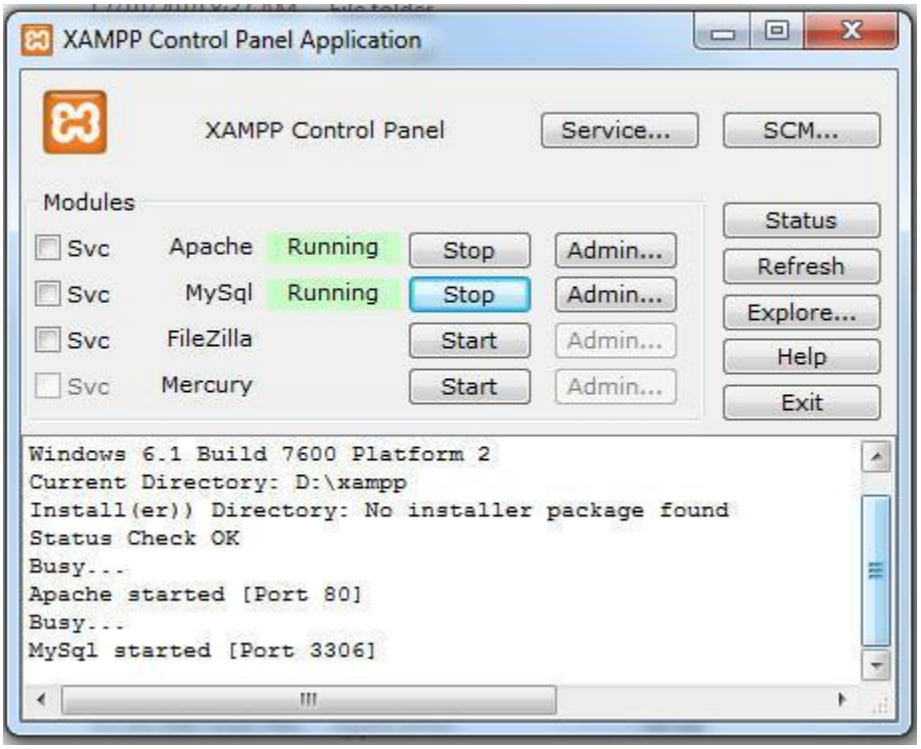


Figure 5.11: XAMPP Control Panel

5.3.2 Connecting the Website to Database

Connection uses the PHP MySQL commands:

- mysqli_connect

- mysqli_query
- mysqli_fetch_array
- mysqli_close

The Plan

- make the connection and select the database
- perform the query on the table
- print out the data
- close the connection

First Up – Connecting to a MySQL database

You need your MySQL *server address* (if the database is on the same server as the web server it will most likely be localhost or 127.0.0.1), *username*, *password* and *database name*. Create a *filenamehere.php* file and open and close the *php* code with tags before the *html*, you can put regular html after it. Open the file in a browser and you should see nothing apart from the title tag, if you see the error the username/password or database name may be wrong. PHP will require that *mysqli* is enabled (it is on most PHP set ups).

Performing a database query

The mysql query is actually performed in the body of the html page, so additional php opening and closing tags will be required. For the query we are going to specify a read of all fields from a given table. The *$query* variable selects all rows in the table. You just need to use the table name.

Put the data on the page

Here we are taking the making a *$result* variable which stores the query we just made above, now we just need to go through all the rows of that query which we need *mysqli_fetch_array* which stores the rows in an array, so now we are storing the *$result* in *mysqli_fetch_array* and passing that into a variable called *$row*.
The *$row* now can be output in a while loop, here the rows of data will be echoed and displayed on the page to when there is no longer any rows of data left, my example uses 4 fields in the table *first_name*, *last_name*, *email* and *city*.

Closing off the connection

Closing the connection will require another set off opening and closing php tags after the closing html tag. It is good practice to close the database connection when the querying is done.

5.4 Designing Website and Mobile Application

5.4.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

5.4.2 Setting up Android Studio

To install Android Studio on Windows, proceed as follows:

1. If you downloaded an .exe file (recommended), double-click to launch it.
2. If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder

and launch studio64.exe(for 64-bit machines) or studio.exe (for 32-bit machines).

3. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

That's it. The following video shows each step of the setup procedure when using the recommended.exe download.

5.4.3 Eclipse IDE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

5.5 App to Website Data Transfer

5.5.1 Retrieving JSON from Firebase

We can read data from our Firebase database by issuing a GET request to its URL endpoint. Let's continue with our blog example from the previous section and read all of our blog post data:

```
curl 'https://docs-examples.firebaseio.com/rest/saving-
data/fireblog/posts.json?print=pretty'
```

A successful request will be indicated by a 200 OK HTTP status code, and the response will contain the data we're retrieving.

Adding URI Parameters

The REST API accepts several query parameters when reading data from our Firebase database. Listed below are the most commonly used parameters. For a full list, refer to the REST API Reference.

Auth request:

The auth request parameter allows access to data protected by Firebase Realtime Database Rules, and is supported by all request types. The argument can either be your Firebase app's secret or an authentication token, as described in the Users in Firebase Projects. In the following example we send a GET request with an auth parameter, where CREDENTIAL is either your Firebase app's secret or an authentication token:

curl 'https://docs-examples.firebaseio.com/rest/saving-data/auth-example.json?auth=CREDENTIAL'

Filtering Data

We can construct queries to filter data based on various factors. To start, you specify how you want your data to be filtered using the orderBy parameter. Then, you combine orderBy with any of the other five parameters: limitToFirst, limitToLast, startAt, endAt, and equalTo.

We can filter data in one of three ways: by **child key**, by **key**, or by **value**. A query starts with one of these parameters, and then must be combined with one or more of the following parameters: startAt, endAt, limitToFirst, limitToLast, or equalTo.

Retrieving Data

Reading Data with GET

We can read data from our Firebase database by issuing a GET request to its URL endpoint. Let's continue with our blog example from the previous section and read all of our blog post data:

curl 'https://docs-examples.firebaseio.com/rest/saving-data/fireblog/posts.json?print=pretty'

A successful request will be indicated by a 200 OK HTTP status code, and the response will contain the data we're retrieving.

Adding URI Parameters

The REST API accepts several query parameters when reading data from our Firebase database. Listed below are the most commonly used parameters. For a full list, refer to the REST API Reference.

auth

The auth request parameter allows access to data protected by Firebase Real time Database Rules, and is supported by all request types. The argument can either be your Firebase app's secret or an authentication token, as described in the Users in Firebase Projects. In the following example we send a GET request with an auth parameter, where CREDENTIAL is either your Firebase app's secret or an authentication token:

curl 'https://docs-examples.firebaseio.com/rest/saving-data/auth-example.json?auth=CREDENTIAL'

5.5.2 Mapping JSON Data to Website

The JSON syntax is like JavaScript's object literal syntax, except that it cannot be assigned to a variable. JSON merely represents the data itself. Since JSON is just a string of text, it needs to be converted to an object to be useful inside JavaScript. Likewise, JavaScript objects need to be converted into strings in order to be used as JSON data.

Two functions for working with JSON are built into JavaScript:

- JSON.parse() - Converts a JSON string into a JavaScript object.
- JSON.stringify() - Converts a JavaScript object into a JSON string.

The following explains how to send data between the browser and server with JSON.

Send JSON Data from the Client Side

1. Create a JavaScript object using the standard or literal syntax.
2. Use JSON.stringify() to convert the JavaScript object into a JSON string.
3. Send the URL-encoded JSON string to the server as part of the HTTP Request. This can be done using the HEAD, GET, or POST method by assigning the JSON string to a variable. It can also be sent as raw text using the POST method, but this may create extra work for you on the server-side.

Receive JSON Data on the Server Side

1. Convert the incoming JSON string to an object using a JSON parser for the language of your choice. At json.org., you'll find JSON parsers for many modern programming languages. The methods available depend upon which parser you are using. See the parser's documentation for details.
2. Do whatever you wish with the object.

Send JSON Data from the Server Side

1. Create a new object for storing the response data.
2. Convert the new object to a string using your JSON parser.
3. Send the JSON string back to the client as the response body (e.g, Response. Write(strJSON), echo $strJSON, out.write(strJSON), etc.).

Receive JSON Data on the Client Side

1. Convert the incoming JSON string to an object using JSON.parse().
2. Do whatever you wish with the object.

CHAPTER 6

RESULT AND DISCUSSION

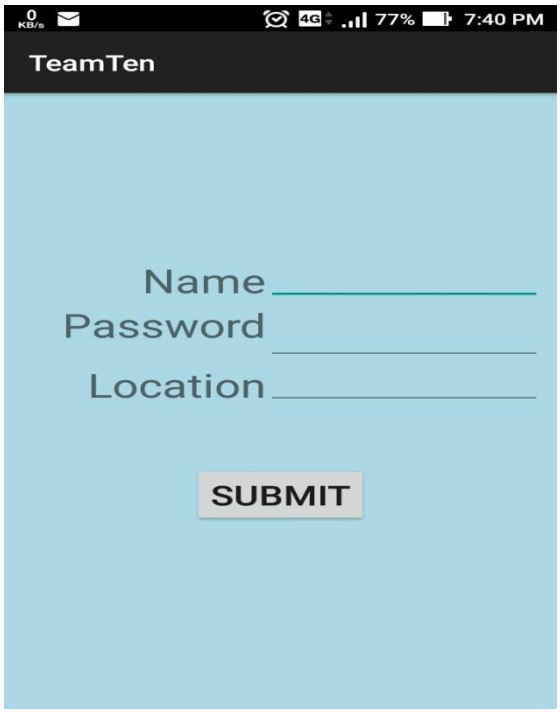


Figure 6.1: Login Screen

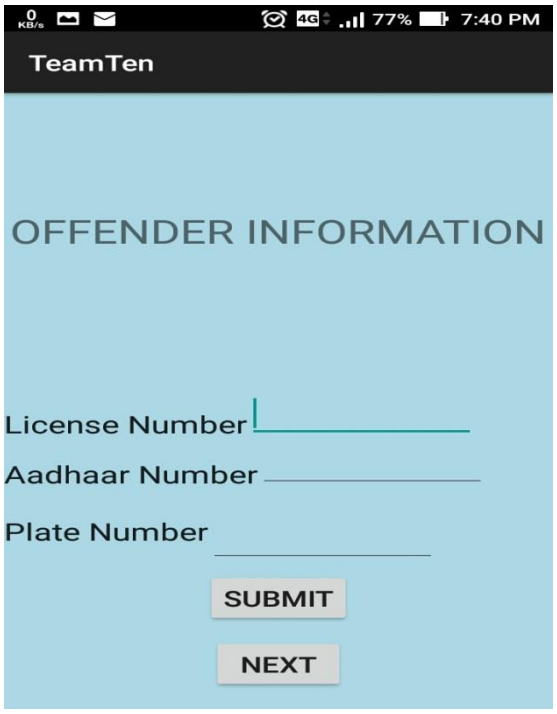


Figure 6.2: Offender Information Screen

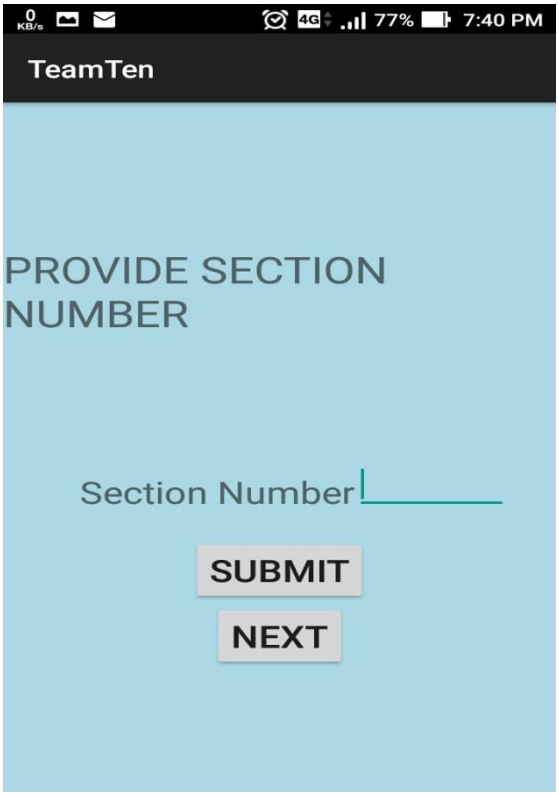


Figure 6.3: Provide Section Number Screen

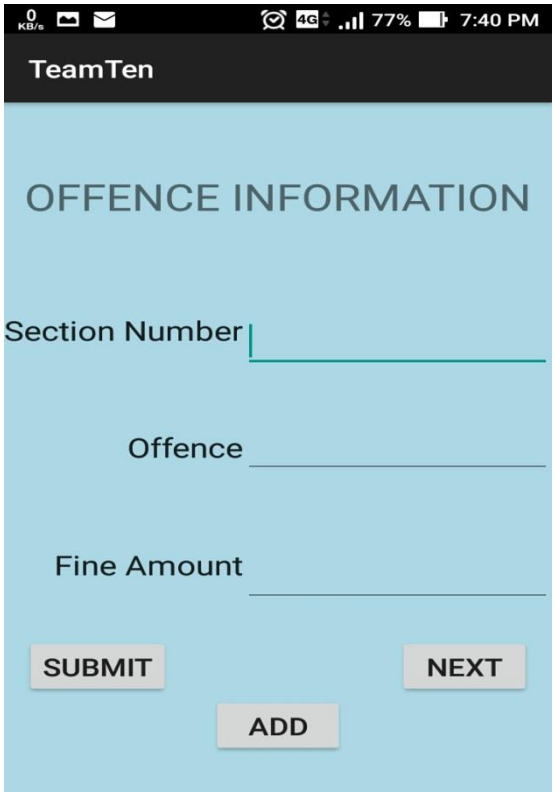


Figure 6.4: Offence Information Screen

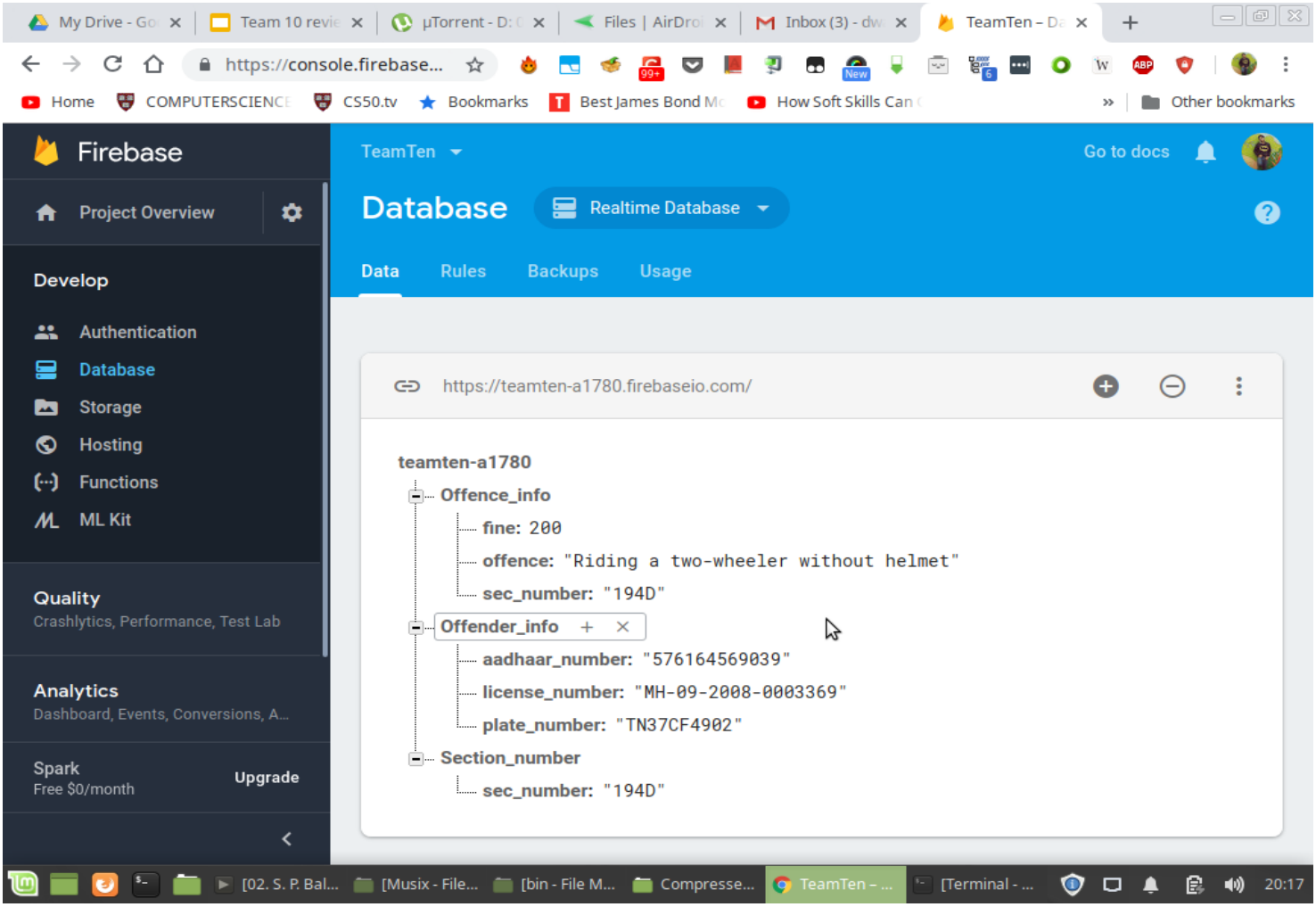


Figure 6.5: RealTime Database - FireBase

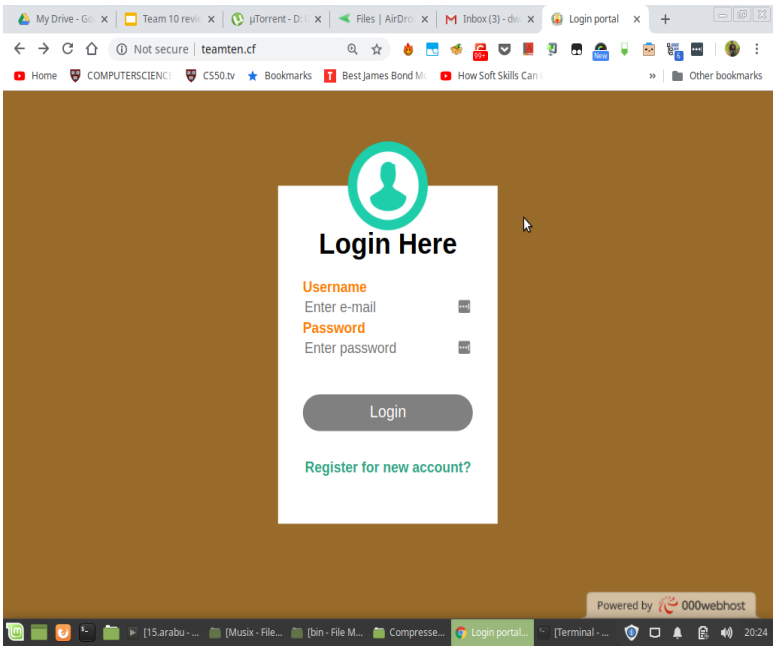


Figure 6.6: Login Screen - Website

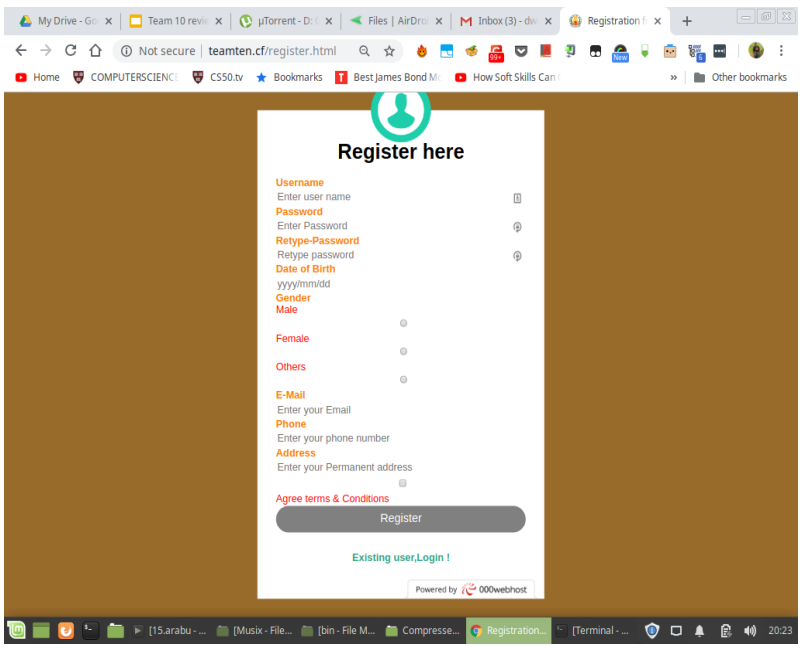


Figure 6.7: Register Screen - Website

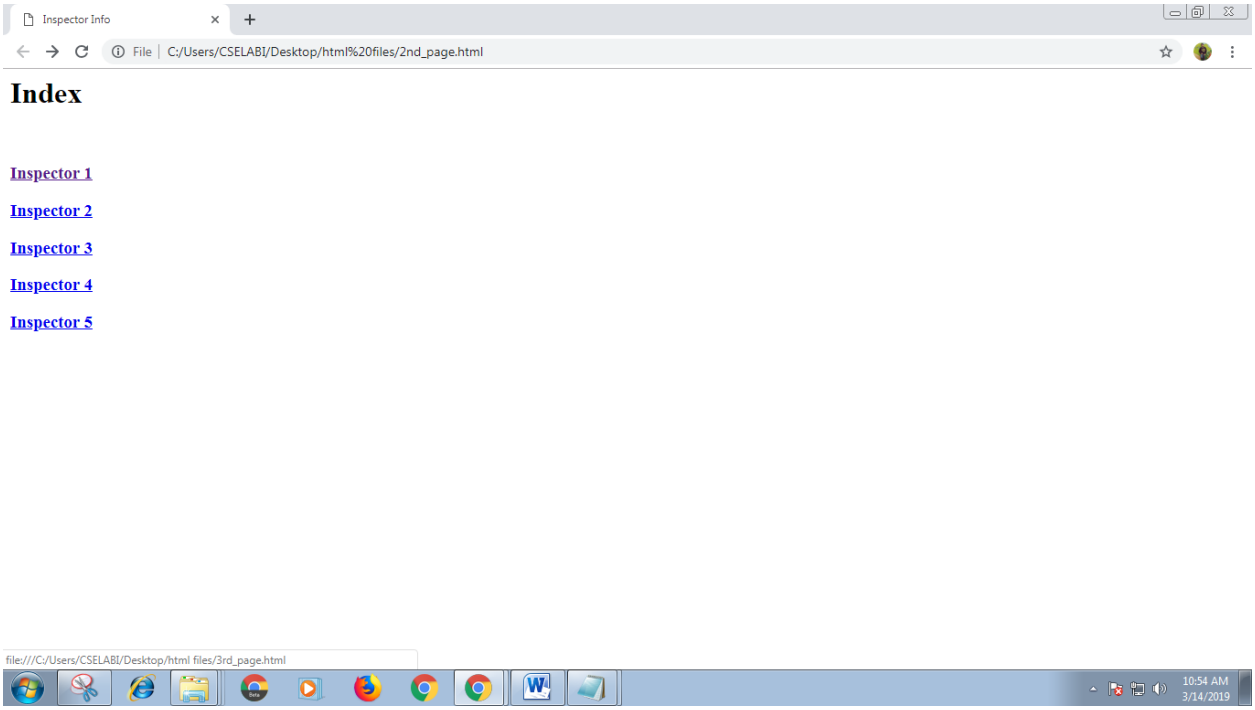


Figure 6.8: Inspector Index Screen

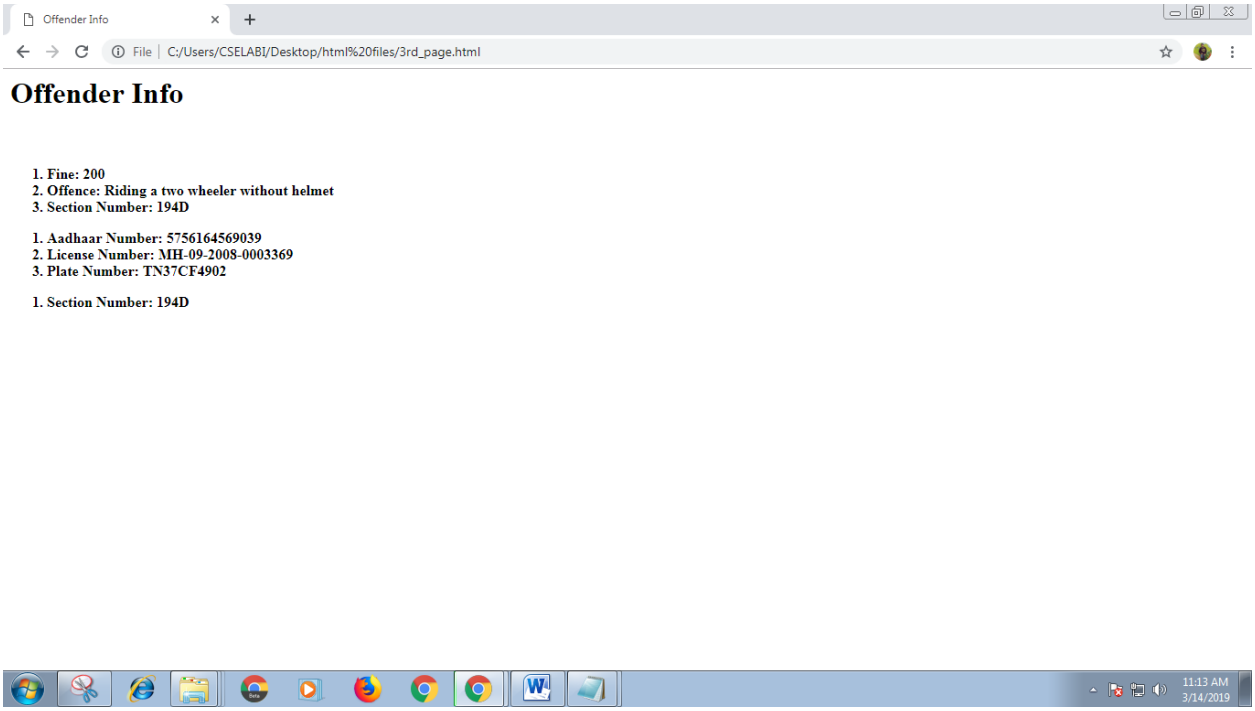


Figure 6.9: Offender Information Screen

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

In this project, we presented an interactive approach to digitalize the traffic fine system. The problem of normal receipt system is replaced by the digital system by means of the android application. Effective solution for acknowledgment from the traffic police is made through an SMS to the offenders. The main aim is to maintain a complete database for the maintenance of the cops and offence details at a large quantity. As the early system has challenges in maintenance of receipted data this proposed project may provide a better enhancement using with the cloud computing.

7.2 Future Enhancement

QR code scanner with respect to the aadhar card will make the process for cobs. Typically, a Smartphone is used as a QR code scanner, displaying the code and converting it to some useful form (such as a standard URL for a website, thereby obviating the need for a user to type it into a web browser). Beyond mere convenience to the cobs, the importance of this capability is that it increases the conversion rate: the chance that contact with the administrator will convert to a faster process. It coaxes interested prospects further down the conversion funnel with little delay or effort, bringing the cobs to the offenders log immediately, where a longer and more targeted sales pitch may lose the offenders interest.

APPENDIX

A.1 Source Code

A.1.1 Mobile Application

MainActivity.java

```
package com.example.dwaki.teamten;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
public class MainActivity extends Activity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        }

        public void next(View view)
        {
        Intent intent = new Intent(this,OffenderInfo.class );
        startActivity(intent);
        }
}
```

OffenceInfo.java

```
package com.example.dwaki.teamten;
import android.app.Activity;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
public class OffenceInfo extends Activity {
        private EditText section;
        private EditText offence;
        private EditText fine;
        private FirebaseDatabase dDatabase;
        private DatabaseReference myRef;
        private Button dbtn;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_offence_info);
        dDatabase = FirebaseDatabase.getInstance();
        myRef = dDatabase.getReference().child("Offence_info");
        section = (EditText)findViewById(R.id.section_edit);
        offence = (EditText)findViewById(R.id.offence_edit);
        fine = (EditText)findViewById(R.id.fine_edit);
        dbtn = (Button)findViewById(R.id.submit_offence);
        dbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

                String sec = section.getText().toString().trim();
                String off = offence.getText().toString().trim();
                String fn = fine.getText().toString().trim();
                myRef.child("section_number").setValue(sec);
```

```
                myRef.child("offence_number").setValue(off);
                myRef.child("fine_number").setValue(fn);
        }
        });
        }
        public void add(View view)
        {
        Intent intent = new Intent(this, SectionEnter.class);
        startActivity(intent);
        }
}
```

OffenderInfo.java

```
package com.example.dwaki.teamten;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
public class OffenderInfo extends Activity {
        private EditText license;
        private EditText aadhaar;
        private EditText plate;
        private FirebaseDatabase dDatabase;
        private DatabaseReference myRef;
        private Button dbtn;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_offender_info);
        dDatabase = FirebaseDatabase.getInstance();
        myRef = dDatabase.getReference().child("Offender_info");
        license = (EditText)findViewById(R.id.license_edit);
        aadhaar = (EditText)findViewById(R.id.aadhaar_edit);
        plate = (EditText)findViewById(R.id.plate_edit);
        dbtn = (Button)findViewById(R.id.submit_offender);
        dbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                String lno = license.getText().toString().trim();
                String ano = aadhaar.getText().toString().trim();
                String pno = plate.getText().toString().trim();
                myRef.child("license_number").setValue(lno);
                myRef.child("aadhaar_number").setValue(ano);
                myRef.child("plate_number").setValue(pno);
        }
        });
        }
        public void next(View view)
        {
        Intent intent = new Intent(this,SectionEnter.class);
        startActivity(intent);
        }
}
```

SectionEnter.java

```
package com.example.dwaki.teamten;
import android.app.Activity;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
public class SectionEnter extends Activity {
        private EditText section;
        private FirebaseDatabase dDatabase;
        private DatabaseReference myRef;
        private Button dbtn;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_section_enter);
        dDatabase = FirebaseDatabase.getInstance();
        myRef = dDatabase.getReference().child("Section_Number_info");
        section = (EditText)findViewById(R.id.sectionno_edit);
        dbtn = (Button)findViewById(R.id.submit_sec);
        dbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                String sec = section.getText().toString().trim();
                myRef.child("section_number").setValue(sec);
        }
        });
        }
        public void next(View view)
        {
        Intent intent = new Intent(this,OffenceInfo.class);
        startActivity(intent);
```

```
        }
}
```

A.1.2 Website

Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Login portal</title>
 <link rel="icon" type="image/ico" href="tnlogo.png" />
<link rel="stylesheet" type="text/css" href="style.css">
<script type="text/javascript"src="jscript.js"></script>
</head>
<body oncontextmenu="return false;">
<div class="box">
<img src="image.png" class="user"><br>
<h1>Login Here</h1>
<form name="loginform" onsubmit="return vfun()">
<p>Username</p>
<input type="text" name="uname"placeholder="Enter e-mail">
<p>Password</p>
<input type="password" name="upswd"placeholder="Enter password">
<div id="errorBox">
</div><br><br>
<input type="submit" name="login"value="Login">
<br>
<a href="register.html"><h4>
Register for new account?</h4></a>
</form>
</div>
</body>
</html>
```

Javascript.js

```
function vfun(){
var uname=document.forms["loginform"]["uname"].value;
var upswd=document.forms["loginform"]["upswd"].value;
if(uname==null || uname==""){
document.getElementById("errorBox").innerHTML =
"Enter the user name";
return false;
}
if(upswd==null || upswd==""){
document.getElementById("errorBox").innerHTML =
"Enter the password";
return false;
}
if (uname != ""&& upswd !=""){
alert("Login successfully");
}
}
function vfun1(){
var uname1=document.forms["register"]["uname"].value;
var email1=document.forms["register"]["email"].value;
var upswd1=document.forms["register"]["upswd1"].value;
var upswd2=document.forms["register"]["upswd2"].value;
if(uname1==null || uname1=="" ){
document.getElementById("errorBox").innerHTML =
"Enter the user name";
return false;
 }
if(email1==null || email1==""){
document.getElementById("errorBox").innerHTML =
"Enter the email";
```

```
return false;
}
if(upswd1==null || upswd1=="" ){
document.getElementById("errorBox").innerHTML =
"Enter the password";
return false;
}
if(upswd2==null || upswd2==""){
document.getElementById("errorBox").innerHTML =
"Enter the password";
return false;}
if(upswd1 != upswd2){
document.getElementById("errorBox").innerHTML =
"Password invalid match";
return false;}
if (uname1 != " && upswd1 != " && upswd2 != " && email1 != " && upswd1 == upswd2)
{
alert("Register successfull");
 }
}
```

Style.css

```
body{
margin:0;
padding:0;
font-family:sans-serif;
background-color:#996b2a;
}
.header{
padding:60px;
background:#1abc9c;
}
```

```
.box{
background:white;
color:red;
top:50%;
left:50%;
position: absolute;
transform:translate(-50%,-50%);
box-sizing:border-box;
padding:30px 30px;
}
.user{
width: 100px;
height: 100px;
position: absolute;
top:-50px;
left: calc(50% - 50px);
}
h1{
margin: 0;
padding: 0 0 20px;
text-align: center;
font size: 20px;
color:black;
}
h4{
text-align: center;
}
p{
color:#FC8004;
margin: 0;
padding:0;
```

```
font-weight: bold;
}
input{
width: 100%;
margin-bottom: 10px;
}
input[type="text"],input[type="password"],input[type="email"]
{
border:none;
margin:2px;
height: 20px;
background: transparent;
color: black;
font-size: 16px;
}
input[type="submit"]
{
border:none;
outline:none;
height:40px;
background: grey;
color:white;
font-size: 18px;
border-radius: 20px;
}
input[type="submit"]:hover
{
cursor:pointer;
background-color:#8b8f96;
}
```

REFERENCES

[1]. Chunnu Khawas, Pritam Shah -Journal International of Computer Applications (0975 – 8887) Volume 179 – No.46, June 2018 Application of Firebase in Android App Development-AStudy.

[2]. Stewart Baker- Making It Work for Everyone: HTML5 and CSS Level 3 for Responsive, Accessible Design on your Library's ,Website Western Oregon University, bakersc@wou.edu.

[3]. Zia Ul Haq , Gul Faraz Khan, Tazar Hussain-A Comprehensive analysis of XML and JSON web technologies

[4]. Suhas Holla, Mahima M Katti- Android Based Mobile Application Development and its SECURITY, International Journal of Computer Trends and Technology- volume3Issue3-2012.

[5]. Arthur Charguéraud Inria & Université de Strasbourg, CNRS, ICube UMR 7357 arthur.chargueraud@inria.fr Alan Schmitt Inria & Univ Rennes, CNRS, IRISAb alan.schmitt@inria.fr Thomas Wood Imperial College London thomas.wood09@imperial.ac.uk- JSExplain: a Double
Debugger for JavaScript.

[6]. P. C. van Oorschot, A. Salehi-Abari, and J. Thorpe.Purely automated attacks on PassPoints-stylegraphical passwords. IEEE Trans. Info. Forensicsand Security, 5(3):393–405, 2010.

[7]. D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson,J. Nicholson, and P. Olivier. Multi-touchauthentication on tabletops. In 28th ACMConference on Human Factor in Computing Systems (CHI), pages 1093–1102, Atlanta, USA, April 2010.

[8]. Ranjan, Alok & Kumar, Rajeev & Dhar, and Joydip. A Comparative Study between Dynamic Web Scripting Languages. 10.1007/978-3-642-27872-3_43. January 2012.

[9]. Mircea Moisoiu, Andrei Negrău, Robert Győrödi, Cornelia Győrödi, George Pecherle. QR Code Scanning app for Mobile Devices. IJCSMC, Vol. 3, Issue. 6, June 2014, pg.334 – 340.

[10]. Botwe, David & G Davis, Joseph. (2015). International Journal of Computer Science and Mobile Computing A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software. 42. 154-165.

[11]. Kishore Baktha. Mobile Application Development: All the Steps and Guidelines for Successful Creation of Mobile App: Case Study. IJCSMC, Vol. 6, Issue. 9, September 2017, pg.15 – 20.

[12]. Shweta Dinesh Bijwe, Prof. P. L. Ramteke. Database in Cloud Computing - Database-as-a Service (DBaas) with its Challenges. IJCSMC, Vol. 4, Issue. 2, February 2015, pg.73 – 79.