

# **DIGITAL MTC PASS USING PWA**

## **A PROJECT REPORT**

*Submitted by*

**DIVYA PRAKASH J**

**113217104030**

**PRAVEENKUMAR RA**

**113217104090**

**SRIRAM B**

**113217104122**

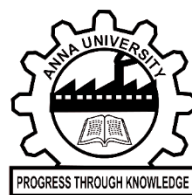
*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**VELAMMAL ENGINEERING COLLEGE  
ANNA UNIVERSITY : CHENNAI 600025  
MARCH 2021**

# **ANNA UNIVERSITY : CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**DIGITAL MTC PASS USING PWA**” is the bonafide work of “**DIVYA PRAKASH J (113217104030), PRAVEENKUMAR RA (113217104090) & SRIRAM B (113217104122)** ” who carried out the project work under my supervision.

### **SIGNATURE**

Dr.B.Murugeshwari

### **HEAD OF THE DEPARTMENT**

Professor

Computer Science and Engineering

Velammal Engineering College

Ambattur – Red Hills Road

Chennai – 600 053.

### **SIGNATURE**

Mrs.T.Subashini

### **SUPERVISOR**

Assistant Professor

Computer Science and Engineering

Velammal Engineering College

Ambattur – Red Hills Road

Chennai – 600 053.

# **ANNA UNIVERSITY: CHENNAI 600 025**

## **VIVA-VOCE EXAMINATION**

The Viva Voce Examination of this project work “**DIGITAL MTC PASS USING PWA**” is a bonafide record of project done at the **Department of Computer Science and Engineering**, Velammal Engineering College during the academic year **2020-21** by

**DIVYA PRAKASH J**

**113217104030**

**PRAVEENKUMAR RA**

**113217104090**

**SRIRAM B**

**113217104122**

of final year Bachelor of Engineering in Computer Science and Engineering  
submitted for the university examination on \_\_\_\_\_

INTERNAL EXAMINER

EXTERNAL EXAMINER

## ACKNOWLEDGEMENT

I wish to acknowledge with thanks to the significant contribution given by the management of our college **Chairman, Dr.M.V.Muthuramalingam**, and our **Chief Executive Officer Thiru. M.V.M. Velmurugan**, for their extensive support.

I would like to thank **Dr. N. Duraipandian, Principal** of Velammal Engineering College, for giving me this opportunity to do this project.

I wish to express my gratitude to our effective **Head of the Department, Dr. B. Murugeshwari**, for her moral support and for her valuable innovative suggestions, constructive interaction, constant encouragement and unending help that have enabled me to complete the project.

I wish to express my indebted humble thanks to our **Project Coordinators, Dr. P.Visu, Mrs. B.Hemalatha and Mrs. D.Jena Catherine Bel**, Department of Computer Science and Engineering for their invaluable guidance in the shaping of this project.

I wish to express my sincere gratitude to my **Internal Guide, Mrs.T.Subashini, Assistant Professor**, Department of Computer Science and Engineering for her guidance, without her this project would not have been possible.

I am grateful to the entire staff members of the Department of Computer Science and Engineering for providing the necessary facilities and to carry out the project. I would especially like to thank my parents for providing me with the unique opportunity to work and for their encouragement and support at all levels. Finally, my heartfelt thanks to **The Almighty** for guiding me throughout life.

## **ABSTRACT**

As digital Technology dominates the current situation, the objective of this app is to provide the latest digital way of using MTC Pass which allows commuters to mark their trip by scanning QR codes to monitor their travel. With the help of the Progressive web app, bus commuters can digitally mark their trip by just scanning a QR code. Instead of carrying an MTC card, commuters can install this app and mark their trip digitally by scanning the QR Code. Also, bus commuters can check the dashboard provided by the app to see the number of trips taken on a respective day with the boarding and destination point. This app maintains a database that stores the commuter's trip details and updates the trip count every day. This app also provides an option to renew the current bus pass. Once renewed, this app starts monitoring the trips till the next renewal. For security purposes, commuters must log in to the app with their mobile number/ bus pass id with their Date of Birth. The login credentials are verified with the details provided at the time of applying for a bus pass.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
	LIST OF ABBREVIATIONS	9
	LIST OF SYMBOLS	9
	LIST OF FIGURES	10
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 History	
	1.1.1 About Website's	11
	1.1.2 History of WWW	11
	1.1.3 Overview	12
	1.2 Classifications	12
	1.2.1 Static Website	12
	1.2.2 Dynamic Website	13
	1.3 Progressive Web App	14
	1.3.1 Introduction	14
	1.3.2 Initial Work / Stage	15
<b>2</b>	<b>LITERATURE SURVEY</b>	
	2.1 Research Works	16
<b>3</b>	<b>SYSTEM ANALYSIS</b>	
	3.1 Web Designing	20
	3.1.1 Hypertext Markup Language	20
	3.1.2 Cascading Style Sheets	23
	3.1.2.1 Bootstrap	26

3.1.3 JQuery	29
3.2 Web Development	32
3.2.1 PHP	32
3.2.2 SQL	36
3.3 Progressive Web App	39
3.3.1 Manifest	39
3.3.2 Web Assembly	39
3.3.3 Data Storage	39
3.3.4 Web Storage	40
3.3.5 Service Workers	40
3.3.6 App Cache	41
<b>4 DESIGN AND IMPLEMENTATION</b>	<b>41</b>
4.1 Methodologies	41
4.2 Three way Architecture	42
4.2.1 Client	42
4.2.2 Server	43
4.2.3 Database	44
4.3 Block Diagram of the entire process	46
4.4 Modules	46
4.5 Module Description	47
4.5.1 Registration with fare calculation	47
4.5.2 Dashboard	47
4.5.3 College/ Commuter Pass	48
4.5.4 Travel as you like pass	49
4.5.5 QR Code	50
4.5.6 Route Detection and Validation	51

	4.5.7 Mockup Payment Gateway	51
	4.5.8 Renewal Process	51
<b>5</b>	<b>CONCLUSION</b>	
	5.1 Conclusion	52
	5.2 Future Work	52
	<b>APPENDICES</b>	
	<b>A</b> Sample Code	53
	<b>B</b> Screenshots	60
	<b>REFERENCES</b>	62



## **LIST OF ABBREVIATIONS**

PWA	Progressive Web App
JS	JavaScript
PHP	Hypertext Preprocessor
HTML	Hyper Text MarkUp Language
CSS	Cascading Style Sheet
SQL	Structured Query Language

## **LIST OF SYMBOLS**

₹	Indian Rupee
↕	Up and Down
QR	Qr Code
🚪	Log - Out
📊	Table Records

## **LIST OF FIGURES**

<b>DIAGRAM</b>	<b>PAGE NO</b>
4.1 Block diagram for Three Tier Architecture	41
4.2 Client Side of Three Tier Architecture	42
4.3 Client Side of Three Tier Architecture	43
4.4 Database	45
4.5 Block diagram of the entire process	45
4.6 Flowchart for Student or Commuter Pass	49
4.7 Flowchart for Travel as you like Pass	50

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 HISTORY**

#### **1.1.1 ABOUT WEBSITE'S**

In computer science, A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server. Notable examples are wikipedia.org, google.com, and amazon.com. Websites are typically dedicated to a particular topic or purpose, such as news, education, commerce, entertainment, or social networking. Hyperlinking between web pages guides the navigation of the site, which often starts with a home page.

#### **1.1.2 HISTORY OF WWW**

The World Wide Web was created in 1990 by the British CERN physicist Tim Berners-Lee. On 30 April 1993, CERN announced that the World Wide Web would be free to use for anyone. Before the introduction of the Hypertext Transfer Protocol, other protocols such as File Transfer Protocol and the gopher protocol were used to retrieve individual files from a server. These protocols offer a simple directory structure which the user navigates and where they choose files to download. Documents were most often presented as plain text files without formatting or were encoded in word processor formats.

### **1.1.3 OVERVIEW**

Websites can be used in various fashions: a personal website, a corporate website for a company, a government website, an organization website, etc. Websites can be the work of an individual, a business or other organization, and are typically dedicated to a particular topic or purpose. Any website can contain a hyperlink to any other website, so the distinction between individual sites, as perceived by the user, can be blurred.

Some websites require user registration or subscription to access content. Examples of subscription websites include many business sites, news websites, academic journal websites, gaming websites, file-sharing websites, message boards, web-based email, social networking websites, websites providing real-time stock market data, as well as sites providing various other services.

## **1.2 CLASSIFICATION OF WEBSITES**

There are 2 types of Websites. They are

1. Static Websites
2. Dynamic Websites

### **1.2.1 STATIC WEBSITE**

A static website is one that has web pages stored on the server in the format that is sent to a client web browser. It is primarily coded in Hypertext Markup Language ; Cascading Style Sheets are used to control appearance beyond basic HTML. Images are commonly used to create the desired appearance and as part of the main content. Audio or video might also be considered "static" content if it plays automatically or is generally non-interactive. This type of website usually displays the same information to all visitors. Similar to handing out a

printed brochure to customers or clients, a static website will generally provide consistent, standard information for an extended period of time. Although the website owner may make updates periodically, it is a manual process to edit the text, photos, and other content and may require basic website design skills and software. Simple forms or marketing examples of websites, such as classic websites, a five-page website or a brochure website are often static websites, because they present pre-defined, static information to the user. This may include information about a company and its products and services through text, photos, animations, audio/video, and navigation menus.

Static websites may still use server side includes as an editing convenience, such as sharing a common menu bar across many pages. As the site's behavior to the reader is still static, this is not considered a dynamic site.

### **1.2.2 DYNAMIC WEBSITE**

A dynamic website is one that changes or customizes itself frequently and automatically. Server-side dynamic pages are generated "on the fly" by computer code that produces the HTML. There are a wide range of software systems, such as CGI, Java Servlets and Java Server Pages, Active Server Pages and ColdFusion that are available to generate dynamic web systems and dynamic sites. Various web application frameworks and web template systems are available for general-use programming languages like Perl, PHP, Python and Ruby to make it faster and easier to create complex dynamic websites.

A site can display the current state of a dialogue between users, monitor a changing situation, or provide information in some way personalized to the

requirements of the individual user. For example, when the front page of a news site is requested, the code running on the web server might combine stored HTML fragments with news stories retrieved from a database or another website via RSS to produce a page that includes the latest information. Dynamic sites can be interactive by using HTML forms, storing and reading back browser cookies, or by creating a series of pages that reflect the previous history of clicks.

## **1.3 PROGRESSIVE WEB APP**

### **1.3.1 INTRODUCTION**

A progressive web application is a type of application software delivered through the web, built using common web technologies including HTML, CSS and JavaScript. It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices. PWA features narrow the gap between user experience in web-based and native applications.

Since a progressive web app is a type of webpage or website known as a web application, they do not require separate bundling or distribution. Developers can just publish the web application online, ensure that it meets baseline "installability requirements", and users will be able to add the application to their home screen. Publishing the app to digital distribution systems like Apple App Store or Google Play is optional.

As of 2021, PWA features are supported to varying degrees by Google Chrome, Apple Safari, Firefox for Android, and Microsoft Edge but not Firefox for desktop. In a wide variety of key performance indicators after PWA implementation, like increased time spent on page, conversions, or revenue.

### **1.3.2 INITIAL WORK / STAGE**

At the launch of the iPhone in 2007, Steve Jobs announced that web apps, developed in HTML5 using AJAX architecture, would be the standard format for iPhone apps. No software development kit was required, and the apps would be fully integrated into the device through the Safari browser engine. This model was later switched for the App Store, as a means of preventing jailbreakers and of appeasing frustrated developers. Firefox released Firefox OS in 2013. It was intended to be an open-source operating system for running web apps as native apps on mobile devices, with Gaia built as its HTML5 interface.

The development of Firefox OS ended in 2016, and the project was completely discontinued in 2017. The core of the system was re-build into a commercial product called KaiOS.

In 2015, designer Frances Berriman and Google Chrome engineer Alex Russell coined the term "progressive web apps" to describe apps taking advantage of new features supported by modern browsers, including service workers and web app manifests, that let users upgrade web apps to progressive web applications in their native operating system .

Google then put significant efforts into promoting PWA development for Android. With Apple's introduction of service worker support for Safari in 2018, PWAs were now supported on the two most commonly-used mobile operating systems, Android and iOS. By 2019, PWAs were available on desktop browsers Microsoft Edge and Google Chrome. In December 2020, Firefox for desktop abandoned implementation of PWAs

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 RESEARCH WORKS**

##### **1. “Smart Bus pass System Using Android”, 25th April 2019**

This project focuses on reducing the workload of both the government as well as passengers where users can apply and renew bus pass through an android application. As well as reduces the documentation process in the renewal and registration of the bus passes in the transport department. To use this application, the user must be registered in the prescribed app, and students who are availing of the bus pass must upload the required documents as their address proof, a photograph of the student. Once the required documents are uploaded, they will be re-directed to the payment page.

Once the documents are verified by the Admin or depot officer, the bus pass with the selected category is sent to the registered mail id. They can get a printout of their bus pass by e-mail.

After they could also carry it in the laminated hard copy format also. This application also collects feedback from the users. Also there is a provision of adding watermarks on the bus pass to additional security against duplicate bus passes.

The available passes that can be bought is monthly pass, senior citizen pass and the student pass. In conclusion, it just provides a online platform to buy and renew bus pass.



## **2. “Smart City Bus Application With QR Code : A Review”, 29th June 2019**

Smart mobile devices becoming common and some stated it as a necessity. This project focuses mainly on QR Payment. The first features that Android bus application lack is that the bus route guide gave only name of places for every bus stop and this do not give the exact desired location.

Next, passengers require information such as the exact location of bus since estimate time arrival can be inconsistent due to traffic jams or random stops at each bus stops. Other than that, passenger required to be on alert for incoming bus because bus can skip to the next bus stop without stopping. This application provides an advanced improvement such as Display bus routes using Google Maps API. Display buses live location on the Google Maps using Global Positioning System (GPS).

Alarm passenger when bus is coming near by invoking notification such as vibration and sound. Display fare based on users' location and destination by calculating number of bus stops. Payment using QR code technology. With the features of the application, passengers be able to plan ahead of their journey.

The application also provides a detailed but simple bus route information resulting in passenger take less time to read and understand route comparing to traditional route guide. Other than that, the applications also provide real-time bus location, expected arrival time, and incoming bus notification for more peace of mind waiting at bus stop.

In conclusion, this application provides an way to pay using qr code. And uses GPS technology to locate the live location of the bus.

### **3. “Bus Pass and Ticket automation System”,8th August 2016**

As technology starts growing we need to update ourselves to current trends and our upcoming generations looking forward to necessary services in one touch. The availing of a bus pass and ticket in a crowded bus makes the process more tedious. Waiting for the conductor to mark the trip on the bus pass, also waiting for the change in a crowded bus is a hectic one.

So this project has an android application through which the bus pass can be bought online. Through this app, a user can buy a bus ticket online and pay for the same through a payment gateway provided by the app. A user can buy a ticket online by just scanning the QR code. It also keeps track of the expiration of the bus pass and notifies the user regarding the expiry of a bus pass.

It makes the registration process simple by just asks the user their aadhar number and phone number. The aadhar number is used for the purpose of verifying the user from the NIC (National Informatics Center) database. Once registered, the user id displayed with two options, one is to avail a new bus pass and the other is to renew the current bus pass. This application also provides an option of an online wallet where the user can load their money in the wallet and pay for the purchase of a bus pass and ticket.

The QR code is pasted on the bus where it contains the bus details, and with the help of the bus details, the bus route is fetched from the database. Once the QR code is scanned and payment is done, the user is provided with a soft copy of the ticket with the QR code which is encrypted. In conclusion, this app provides a facility to buy tickets by scanning a QR code.

#### **4. Efficient Bus Pass Generation and Authentication using QR Code, 20th June 2020**

The main aim of the project is to effective bus pass generation and authentication using QR code. In this project, the entire bus pass generation is made as an online application. So one can easily apply and renew their bus pass in a simple and easy manner. This application will be useful for students as well as regular passengers. Both the admins and passengers have login pages. Once the user has registered, a unique user id will be generated for the user. The user can use the user id to login into the application. If a user is a student, then the details will be sent to the school admin for verification. Once the admin verifies the details, the bus pass will be availed for the student. Bus conductor or ticker checker can know the details about the bus pass by just scanning through their device .The role of school admin is introduced to decrease the work of admin and avoid duplicate pass users. If the pass is valid the it will provide all the route and basic information about the passenger. If the pass is not valid for duplicate it will prompt as not valid. So it is easy to prevent fraudulent activities by using this application. This system consists of application form in website where the user can apply for pass in online and renew it. Every user need to upload some documents for verification purpose. There is a new user involved called School Admin to reduce the workload of the bus conductor or ticket checker. This application provides user- friendly features like the user can login, apply for pass, apply for renewal, view status of apply, logout. The admin can login, view pass apply, view user, view renewal apply, add user, add or remove route of bus , and log out.

In conclusion, this application provides an online way to apply and renew bus pass and stores the ticket details.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 WEB DESIGNING**

##### **3.1.1 HYPERTEXT MARKUP LANGUAGE**

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<h1>` and `<p>` directly introduce content into the page. Other tags such as `<div>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the

look and layout of content. The World Wide Web Consortium, former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed "some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called, first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language -based documentation format at CERN. Eleven of these elements still exist in HTML 4.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional \*use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS operating system: these formatting commands

were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force with the mid-1993 publication of the first proposal for an HTML specification, the "Hypertext Markup Language " Internet Draft by Berners-Lee and Dan Connolly, which included an SGML Document type definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ ", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms.

After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based. Further development under the auspices of the IETF was stalled by competing interests. The HTML specifications have been maintained, with input from commercial software vendors, by the World Wide Web Consortium . However, in 2000, HTML also became an international standard . HTML 4.01 was published in late 1999, with further errata published through 2001. In 2004, development began on HTML5 in the Web Hypertext Application Technology Working Group, which became a joint deliverable with the W3C in 2008, and completed and standardized on 28 October 2014.

### 3.1.2 CASCADING STYLE SHEETS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice, and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium . Internet media type text/css is registered for use with CSS by RFC 2318 . The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL. CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet

consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters, hyphens, and underscores. A class may apply to any number of instances of any elements. An ID may only be applied to a single element.

Pseudo-classes are used in CSS selectors to permit formatting based on information that is not contained in the document tree. One example of a widely used pseudo-class is `:hover`, which identifies content only when the user "points to" the visible element, usually by holding the mouse cursor over it. It is appended to a selector as in `div:hover` or `a:hover`. A pseudo-class classifies document elements, such as `:hover` or `:focus`, whereas a pseudo-element makes a selection that may consist of partial elements, such as `:first-child` or `:last-child`, whereas a pseudo-element makes a selection that may consist of partial elements, such as `:first-letter` or `:first-line`. Selectors may be combined in many ways to achieve great specificity and flexibility. Multiple selectors may be joined in a spaced list to specify elements by location, element type, id, class, or any combination thereof. The order of the selectors is important. For example, `div .myClass`

Each web browser uses a layout engine to render web pages, and support for CSS functionality is not consistent between them. Because browsers do not parse CSS perfectly, multiple coding techniques have been developed to target specific browsers with workarounds. Adoption of new functionality in CSS can be hindered by lack of support in major browsers. For example, Internet Explorer was slow to add support for many CSS 3 features, which slowed adoption of those features and damaged the browser's reputation among developers. In order to ensure a consistent experience for their users, web developers often test their sites across multiple operating systems, browsers, and browser versions, increasing



development time and complexity. Tools such as BrowserStack have been built to reduce the complexity of maintaining these environments.

In addition to these testing tools, many sites maintain lists of browser support for specific CSS properties, including [CanIUse](#) and the Mozilla Developer Network. Additionally, the CSS 3 defines feature queries, which provide an `@supports` directive that will allow developers to target browsers with support for certain functionality directly within their CSS. CSS that is not supported by older browsers can also sometimes be patched in using JavaScript polyfills, which are pieces of JavaScript code designed to make browsers behave consistently. These workarounds—and the need to support fallback functionality—can add complexity to development projects, and consequently, companies frequently define a list of browser versions that they will and will not support.

As websites adopt newer code standards that are incompatible with older browsers, these browsers can be cut off from accessing many of the resources on the web . Many of the most popular sites on the internet are not just visually degraded on older browsers due to poor CSS support, but do not work at all, in large part due to the evolution of JavaScript and other web technologies.

Selectors are unable to ascend: CSS currently offers no way to select a parent or ancestor of an element that satisfies certain criteria. CSS Selectors Level 4, which is still in Working Draft status, proposes such a selector, but only as part of the complete "snapshot" selector profile, not the fast "live" profile used in dynamic CSS styling. A more advanced selector scheme would enable more sophisticated style sheets. The major reasons for the CSS Working Group previously rejecting proposals for parent selectors are related to browser performance and incremental rendering issues.

Cannot explicitly declare new scope independently of position: Scoping rules for properties such as z-index look for the closest parent element with a position:absolute or position:relative attribute.

This odd coupling has undesired effects. For example, it is impossible to avoid declaring a new scope when one is forced to adjust an element's position, preventing one from using the desired scope of a parent element.

Pseudo-class dynamic behavior not controllable: CSS implements pseudo-classes that allow a degree of user feedback by conditional application of alternate styles. One CSS pseudo-class, ":", is dynamic and has potential for misuse, but CSS has no ability for a client to disable it or limit its effects .

Cannot name rules: There is no way to name a CSS rule, which would allow client-side scripts to refer to the rule even if its selector changes.

Cannot include styles from a rule into another rule: CSS styles often must be duplicated in several rules to achieve a desired effect, causing additional maintenance and requiring more thorough testing. Some new CSS features were proposed to solve this, but were abandoned afterwards.

Cannot target specific text without altering markup: Besides the pseudo-element, one cannot target specific ranges of text without needing to utilize place-holder elements.

### **3.1.2.1 BOOTSTRAP**

Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms.

Bootstrap also supports JavaScript extensions. It is the most popular CSS Framework for developing responsive and mobile-first websites. Bootstrap 4 is the newest version of Bootstrap.

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

Bootstrap 4 has significant changes includes Major rewrite of the code , Replacing Less with Sass , Addition of Reboot a collection of element-specific CSS changes in a single file based on Normalize , Dropping support for IE8, IE9, and iOS 6 , CSS Flexible Box support , Adding navigation customization options , Adding responsive spacing and sizing utilities , Switching from the pixels unit in CSS to

root ems , Increasing global font size from 14px to 16px for enhanced readability , Dropping the panel, thumbnail, pager, and well components , Dropping the Glyphicons icon font , Huge number[quantify] of utility classes , Improved form styling, buttons, drop-down menus, media objects and image classes

Bootstrap 4 supports the latest versions of Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE10 and the latest Firefox Extended Support Release (ESR) .

Major changes includes Dropping jQuery in favor of vanilla JavaScript , Rewriting the grid to support columns placed outside of rows and responsive gutters , Migrating the documentation from , Jekyll to Hugo , Dropping support for IE10 and IE11 , Moving testing infrastructure from QUnit to Jasmine , Adding custom set of SVG icons , Adding CSS custom properties , Improved API , Enhanced grid system , Improved customizing docs , Updated forms and RTL support.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns. A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

### 3.1.3 JQUERY

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having at least 3 to 4 times more usage than any other JavaScript library.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine, created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API. Later, this style has been enhanced with a deeper algorithm-data fusion in an heir of jQuery, the D3.js framework.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

jQuery, at its core, is a Document Object Model manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For

example, jQuery can be used for finding an element in the document with a certain property, changing one or more of its attributes, or making it respond to an event .

jQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. jQuery also aims to incorporate other highly used JavaScript functionality .

Separation of JavaScript and HTML: The jQuery library provides simple syntax for adding event handlers to the DOM using JavaScript, rather than adding HTML event attributes to call JavaScript functions. Thus, it encourages developers to completely separate JavaScript code from HTML markup.

Brevity and clarity: jQuery promotes brevity and clarity with features like "chainable" functions and shorthand function names.

Elimination of cross-browser incompatibilities: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another. Like other JavaScript toolkits, jQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.

DOM element selections using the multi-browser open source selector engine Sizzle, a spin-off of the jQuery project. DOM manipulation based on CSS selectors that uses elements' names and attributes, such as id and class, as criteria to select nodes in the DOM

jQuery provides two kinds of functions, static utility functions and jQuery object methods. Each has its own usage style.

Both are accessed through jQuery's main identifier: jQuery. This identifier has an alias named \$. All functions can be accessed through either of these two names.

The jQuery function is a factory for creating a jQuery object that represents one or more DOM nodes. jQuery objects have methods to manipulate these nodes. These methods are chainable as each method also returns a jQuery object.

Access to and manipulation of multiple DOM nodes in jQuery typically begins with calling the \$ function with a CSS selector string. This returns a jQuery object referencing all the matching elements in the HTML page. \$, for example, returns a jQuery object with all the div elements of class test. This node set can be manipulated by calling methods on the returned jQuery object.

These are utility functions and do not directly act upon a jQuery object. They are accessed as static methods on the jQuery or \$ identifier. For example, \$.ajax is a static method.

jQuery provides a \$.noConflict function, which relinquishes control of the \$ name. This is useful if jQuery is used on a Web page also linking another library that demands the \$ symbol as its identifier. In no-conflict mode, developers can use jQuery as a replacement for \$ without losing functionality.

jQuery's architecture allows developers to create plug-in code to extend its function. There are thousands of jQuery plug-ins available on the Web that cover a range of functions, such as Ajax helpers, Web services, data grids, dynamic lists, XML and XSLT tools, drag and drop, events, cookie handling, and modal windows.

## **3.2 WEB DEVELOPMENT**

### **3.2.1 PHP**

PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface executable. On a web server, the result of the interpreted and executed PHP code which may be any type of data, such as generated HTML or binary image data would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command-line interface .

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. PHP/FI could be used to build simple, dynamic web applications. To accelerate bug reporting and improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools version 1.0" on the Usenet discussion group `comp.infosystems.www.authoring.cgi` on June 8, 1995. This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax



resembled that of Perl, but was simpler, more limited and less consistent. A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

The fact that PHP was not originally designed, but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping", while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

However, as no requirement exists for PHP code to be embedded in HTML, the simplest version of Hello, World! may be written like this, with the closing tag `</?>` omitted as preferred in files containing pure PHP code. The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP, although non-PHP text is still subject to control structures described in PHP code. The most common delimiters are `<?>` to open and `?>` to close PHP sections. The shortened form `<?>` also exists. This short delimiter makes script files less portable, since support for them can be disabled in the local PHP configuration and it is therefore discouraged; there is no recommendation against the echo short tag `<?>`. Prior to PHP 5.4.0, this short syntax for echo only works with the `short_open_tag` configuration setting enabled, while for PHP 5.4.0 and later it is always available. The purpose of all these delimiters is to separate PHP code from non-PHP content, such as JavaScript code or HTML markup.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. PHP 5 introduced type declarations that allows functions to force their parameters to be objects of a specific class, arrays, interfaces or callback

functions. However, before PHP 7, type declarations could not be used with scalar types such as integer or string. PHP treats newlines as whitespace in the manner of a free-form language, and statements are terminated by a semicolon. PHP has three types of comment syntax: `/* */` marks block and inline comments; `//` or `#` are used for one-line comments. The `echo` statement is one of several facilities PHP provides to output text.

PHP includes various free and open-source libraries in its source distribution, or uses them in resulting PHP binary builds. PHP is fundamentally an Internet-aware system with built-in modules for accessing File Transfer Protocol servers and many database servers, including PostgreSQL, MySQL, Microsoft SQL Server and SQLite, LDAP servers, and others. Numerous functions familiar to C programmers, such as those in the `stdio` family, are available in standard PHP builds.

PHP allows developers to write extensions in C to add functionality to the PHP language. PHP extensions can be compiled statically into PHP or loaded dynamically at runtime. Numerous extensions have been written to add support for the Windows API, process management on Unix-like operating systems, multibyte strings, `cURL`, and several popular compression formats.

Other PHP features made available through extensions include integration with IRC, dynamic generation of images and Adobe Flash content, PHP Data Objects as an abstraction layer used for accessing databases, and even speech synthesis.

Some of the language's core functions, such as those dealing with strings and arrays, are also implemented as extensions. The PHP Extension Community Library project is a repository for extensions to the PHP language.

There are two primary ways for adding support for PHP to a web server – as a native web server module, or as a CGI executable. PHP has a direct module interface called Server Application Programming Interface, which is supported by many web servers including Apache HTTP Server, Microsoft IIS, Netscape and iPlanet.

Some other web servers, such as OmniHTTPd, support the Internet Server Application Programming Interface, which is a Microsoft's web server module interface. If PHP has no module support for a web server, it can always be used as a Common Gateway Interface or FastCGI processor; in that case, the web server is configured to use PHP's CGI executable to process all requests to PHP files.

PHP-FPM is an alternative FastCGI implementation for PHP, bundled with the official PHP distribution since version 5.3.3. When compared to the older FastCGI implementation, it contains some additional features, mostly useful for heavily loaded web servers.

When using PHP for command-line scripting, a PHP command-line interface executable is needed. PHP supports a CLI server application programming interface since PHP 4.3.0. The main focus of this SAPI is developing shell applications using PHP. There are quite a few differences between the CLI SAPI and other SAPIs, although they do share many of the same behaviors.

PHP has a direct module interface called SAPI for different web servers; in case of PHP 5 and Apache 2.0 on Windows, it is provided in form of a DLL file called, which is a module that, among other functions, provides an interface between PHP and the web server, implemented in a form that the server understands. This form is what is known as a SAPI.

There are different kinds of SAPIs for various web server extensions. For example, in addition to those listed above, other SAPIs for the PHP language include the Common Gateway Interface and command-line interface .

PHP can also be used for writing desktop graphical user interface applications, by using the PHP-GTK extension. PHP-GTK is not included in the official PHP distribution, When PHP is installed and used in cloud environments, software development kits are provided for using cloud-specific features.

### **3.2.2 SQL**

A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems . In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a

component of the LAMP web application software stack, which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, or a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".

MySQL was created by a Swedish company, MySQL AB, founded by Swedes David Axmark, Allan Larsson and Finland Swede Michael "Monty" Widenius.

Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the mSQL antecedent.

Oracle MySQL offers a high availability solution with a mix of tools including the MySQL router and the MySQL shell. They are based on Group Replication, open source tools.

MySQL can also be run on cloud computing platforms such as Microsoft Azure, Amazon EC2, . Some common deployment models for MySQL on the cloud are: Virtual machine image, MySQL as a service, User interfaces , Graphical user interfaces

A graphical user interface is a type of interface that allows users to interact with electronic devices or programs through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. MySQL Workbench is the integrated environment for MySQL. It was developed by MySQL AB, and enables users to graphically administer MySQL databases and visually design database structures. MySQL shell is a tool for interactive use and administration of the MySQL database. It supports JavaScript, Python or SQL modes and it can be used for administration and access purposes.

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for .NET/CLI Languages, and the JDBC driver for Java.

### **3.3 PROGRESSIVE WEB APP**

#### **3.3.1 MANIFEST**

The web app manifest is a W3C specification defining a JSON-based manifest. The web app manifest provides information about a web application in a JSON text file, necessary for the web app to be downloaded and be presented to the user similarly to a native app (e.g., be installed on the homescreen of a device, providing users with quicker access and a richer experience). PWA manifests include its name, author, icon(s), version, description, and list of all the necessary resources.

#### **3.3.2 WEB ASSEMBLY**

WebAssembly allows precompiled code to run in a web browser, at near-native speed. Thus, libraries written in languages such as C can be added to web apps. Due to the cost of passing data from JavaScript to WebAssembly, near-term uses will be mainly number-crunching, rather than whole applications.

#### **3.3.3 DATA STORAGE**

Progressive Web App execution contexts get unloaded whenever possible, so progressive web apps need to store majority of long-term internal state in one of the following manners

### **3.3.4 WEB STORAGE**

Web Storage is a W3C standard API that enables key-value storage in modern browsers. The API consists of two objects, `sessionStorage` and `localStorage`. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

### **3.3.5 SERVICE WORKERS**

A service worker is a web worker that implements a programmable network proxy that can respond to web/HTTP requests of the main document. It is able to check the availability of a remote server and to cache content when that server is available, and serve that content later to the document. Service workers, like any other web workers, work separately from the main document context. Service workers can handle push notifications and synchronize data in the background, cache or retrieve resource requests, intercept network requests and receive centralized updates independently of the document that registered them, even when that document is not loaded.

Service workers go through a three-step lifecycle of Registration, Installation and Activation. Registration involves telling the browser the location of the service worker in preparation for installation. Installation occurs when there is no service



worker installed in the browser for the webapp, or if there is an update to the service worker. The lifecycle also helps maintain consistency when switching among versions of service worker since only a single service worker can be active for a domain. Indexed Database API can be used with a wrapper library providing additional constructs around it.

### **3.3.6 APP CACHE**

Application Cache is an earlier technology that allowed the application to cache content in advance for later use when the device is offline. It works adequately for single-page applications which it was designed for, but fails in problematic ways for multi-page applications like wikis. As of May 2019, the technology is supported by major browsers and in use for years by some sites, but it is already deprecated in favor of service workers and will eventually be removed.

## **CHAPTER 4**

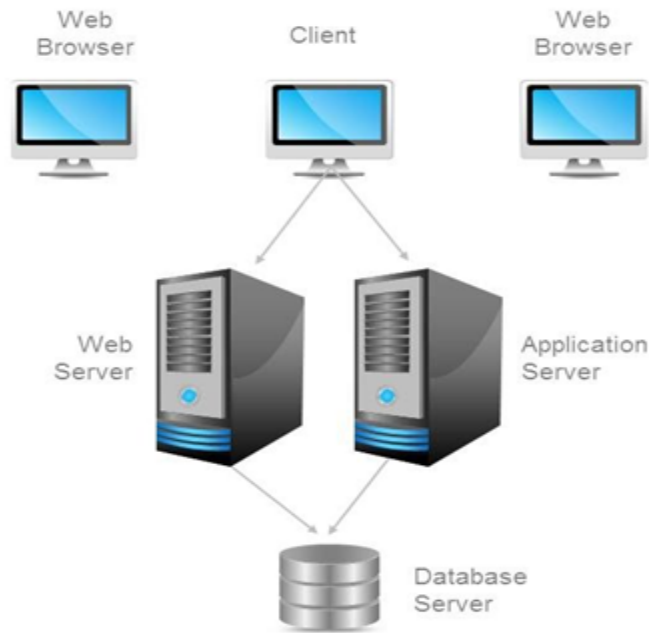
### **DESIGN AND IMPLEMENTATION**

#### **4.1 METHODOLOGIES**

Digital MTC pass is a progressive web application. The main idea of digital MTC pass is to replace the MTC card. This process is achieved by the most famous 3-way architecture. The 3-way architecture includes

1. Client
2. Server
3. Database

## 4.2 THREE WAY ARCHITECTURE



**Fig 4.1** Block diagram of Three Tier Architecture

### 4.2.1 CLIENT

Client-side refers to operations that are performed by the client in a client–server relationship in a computer network.

Typically, a client is a computer application, such as a web browser, that runs on a user's local computer, smartphone, or other device, and connects to a server as necessary. Operations may be performed client-side because they require access to information or functionality that is available on the client but not on the server, because the user needs to observe the operations or provide input, or because the server lacks the processing power to perform the operations in a timely manner for all of the clients it serves.

When the server serves data in a commonly used manner, for example according to standard protocols such as HTTP or FTP, users may have their choice of a number of client programs. In the case of more specialized applications, programmers may write their own server, client, and communications protocol which can only be used with one another.

Programs that run on a user's local computer without ever sending or receiving data over a network are not considered clients, and so the operations of such programs would not be termed client-side operations.



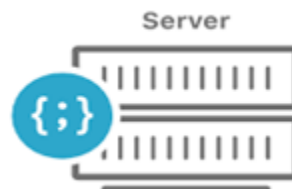
**Fig 4.2:** Client Side of Three tier architecture

### **4.2.2 SERVER**

Server-side refers to operations that are performed by the server in a client–server relationship in a computer network.

A server is a computer application, such as a web server, that runs on a remote server, reachable from a user's local computer, smartphone, or other device. Operations may be performed server-side because they require access to information or functionality that is not available on the client, or because performing such operations on the client side would be slow, unreliable, or insecure. Client and server programs may be commonly available ones such as free or commercial web servers and web browsers, communicating with each other using standardized protocols. Or programmers may write their own server, client, and communications protocol which can only be used with one another. In the case

of distributed computing projects such as SETI@home and the Great Internet Mersenne Prime Search, while the bulk of the operations occur on the client side, the servers are responsible for coordinating the clients, sending them data to analyze, receiving and storing results, providing reporting functionality to project administrators, etc. In the case of an Internet-dependent user application like Google Earth, while querying and display of map data takes place on the client side, the server is responsible for permanent storage of map data, resolving user queries into map data to be returned to the client, etc.



**ig 4.3:** Server Side of Three tier architecture

### 4.2.3 DATABASE

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex, they are often developed using formal design and modeling techniques.

The database management system is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

Computer scientists may classify database-management systems according to the database models that they support. Relational databases became dominant in the

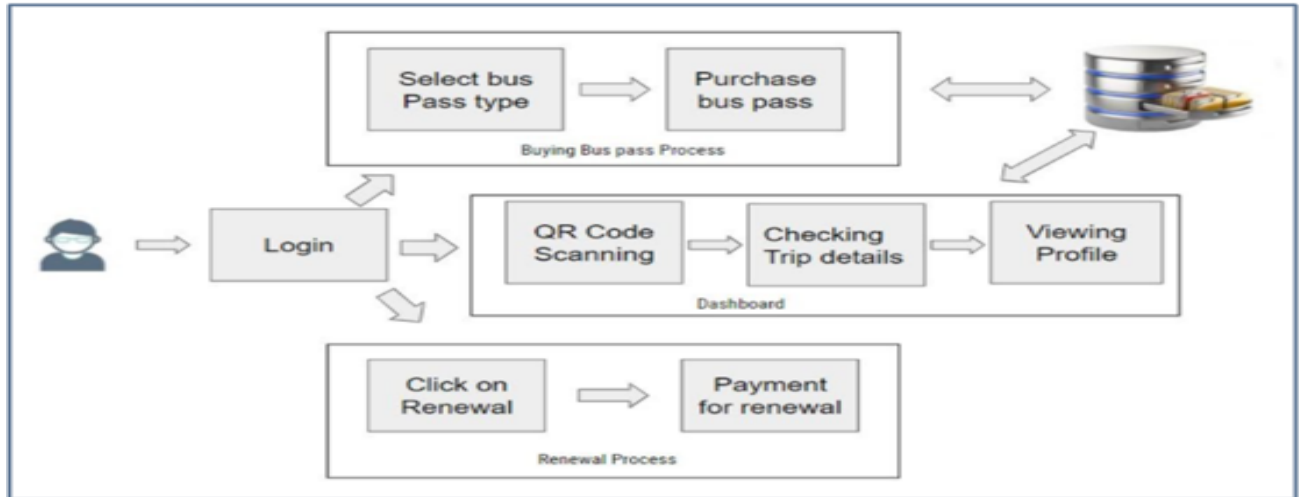
1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, referred to as NoSQL because they use different query languages.

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database . The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized. Outside the world of professional information technology, the term database is often used to refer to any collection of related data as size and usage requirements typically necessitate use of a database management system.



**Fig 4.4 Database**

### 4.3 BLOCK DIAGRAM OF THE ENTIRE PROCESS



**Fig 5.5** Block diagram of the entire process

### 5.4 MODULES

- A. Registration with Fare Calculation
- B. Dashboard
- C. College or Commuter Pass
- D. Travel as you like Pass (50, 1000)
- E. QR Code
- F. Route Detection & Checking
- G. Mockup Payment Gateway
- H. Renewal Process

## **4.5 MODULE DESCRIPTION**

A module is a separate unit of software or hardware. Characteristics of modular components include portability and interoperability which allows them to function in another system with the components of other systems.

### **4.5.1 Registration with Fare Calculation**

A commuter or a college student can register their pass in the registration portal. A username, the Mail address and phone number and their type of Pass is connected. MTC pass include 2 types:

1. One way
2. Two way

Once the user enters the details in the registration portal the site automatically calculates the amount to be paid for pass. The fare is determined by the number of buses, Number of Stages and Type of user (college or Commuter).

This fare calculation has 3 test cases

1. One Way - A direct route where the user can reach is Destination in a single bus and in a Single route
  - a. The Fare is calculated based on the number of stages in between the Starting Point and Destination
  - b. If this route is covered by any other different Bus, then the fare is the one maximum number of stages
2. Two Way – Here the commuter needs to take two buses in order to reach his Destination Point

- a. The Fare is the calculated sum of the number of stages in between the Starting Point and Destination of Two buses
- b. If any one of the routes is covered by any other buses, then the bus which has maximum stages to reach the Destination is taken into account

Once the fare is calculated then the Portal redirects to a Payment Gateway where the users can choose any one of the Payment methods to Pay the fare. A Mail will be sent as Acknowledgement with receipt to the user.

#### **4.5.2 Dashboard**

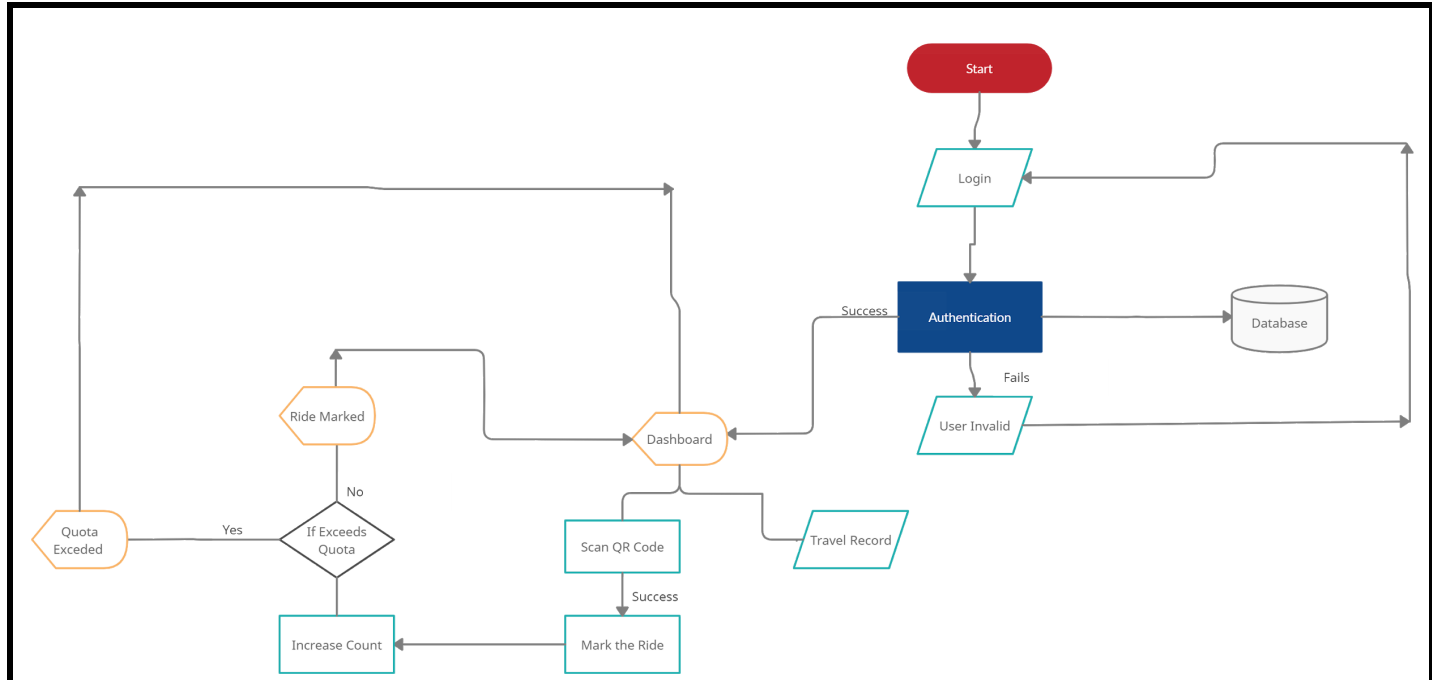
Every user login to their dashboard with Name and Phone Number. In Dashboard the user can scan the QR code to mark their ride. They can also see their number rides they took which differ from each user

#### **4.5.3 College or Commuter Pass**

This pass is issued for pre – registered users only. They are two Different type of these passes

1. One – Way : User can scan up to 2 count which they ride twice up and down in the bus which travel to their route
2. Two – Way: User can scan up to 4 count which they ride twice up and down in different bus which travel to their respective routes





**Fig 4.6 Flowchart for Student or Commuter Pass**

#### 4.5.4 Travel as you like pass

This Pass can be bought any time, Usually in these types of passes user records are not stored. They come with 2 Types

1. Day Pass: User have unlimited count to any destination valid for 24 hours
2. Month Pass: User have unlimited count to any destination valid for 30 days

Users can buy this Pass , by clicking Buy Pass at any time. This redirect to payment portal where user can pay through Credit / Debit card or Online Payment

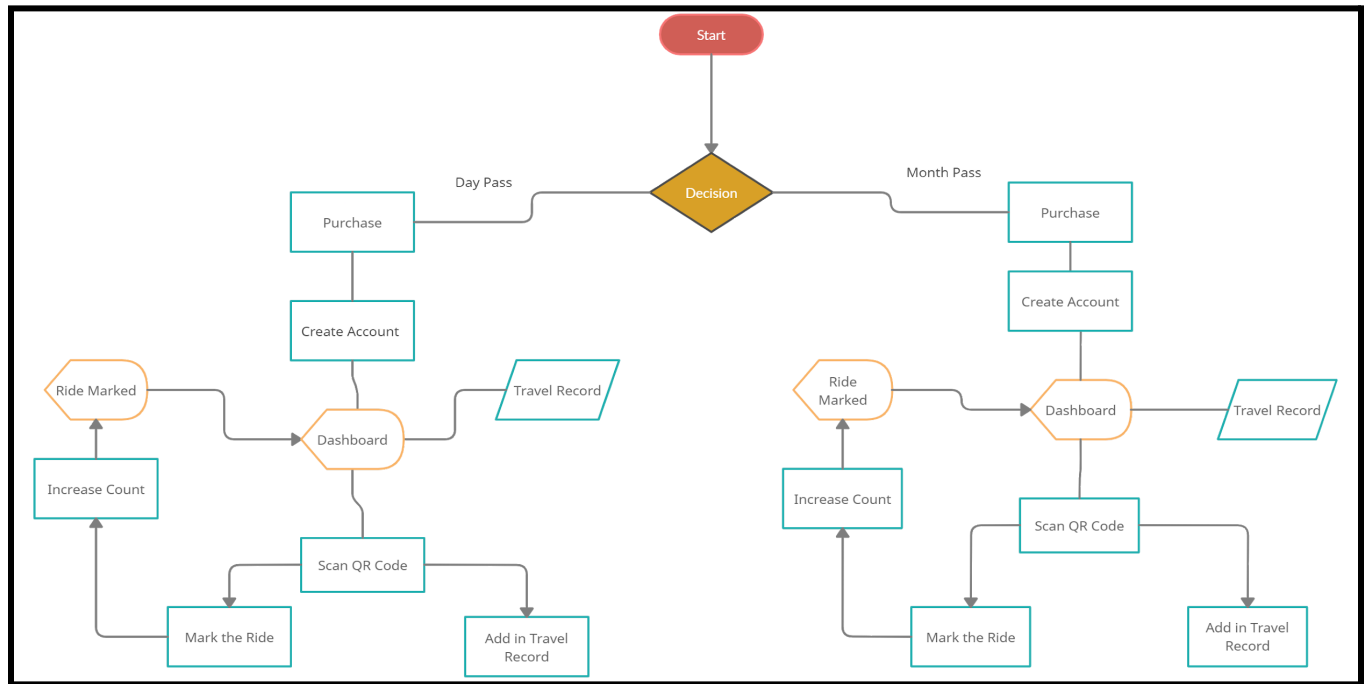


Fig4.7 : Flowchart for Travel as you Like Pass

#### 4.5.5 QR Code

Every bus commuter must scan the QR code attached to the MTC bus. Once scanned, the trip count gets updated in the database for monitoring purposes. Updation of count plays an important role, because it is used to verify whether the user has exceeded the quota.

Every scan is stored in the Database the users can see it by clicking the Table button. The minute the user scanned, the time of scanning, Date, Starting point and Destination Point are stored

A pure javascript QR code reading library. This library takes in raw images and will locate, extract and parse any QR code found within. This allows it to just as easily scan a frontend webcam stream, then data scanned is sent for validation

#### **4.5.6 Route Detection and Validation**

In the previous module we saw that QR code data is sent for validation. This QR code validation differs from different types of Users.

It checks whether it is a QR code Designed for Bus or not. The data received is the Bus number. When Scanned this number returns an array of Stages. If the Starting point and Destination of the user matches with the array of Stages then the user ride is marked.

#### **4.5.7 Mockup Payment Gateway**

This gateway resembles the real payment , but instead real money a mockup money is transferred. The Fare , Name and Mail address is displayed during this payment. A Mockup Receipt is sent to users mail address , with help of inbuilt smtp support in PHP.

#### **4.5.8 Renewal Process**

Renewal Process in this app is applicable only for College and Seasonal pass users. Users Can Renew by clicking on the renew button that appears only at 1<sup>st</sup> of every month till 10<sup>th</sup> of every month. The Renewal process as following testcase

1. Users who renew in between 1<sup>st</sup> and 10<sup>th</sup> of every month. After renewing their pass validity will be updated to next month till 10<sup>th</sup>
2. Users who fail to renew the pass, their Scan to Mark will be revoked temporarily until they renew

Onclick the renew button user will be redirected to payment gateway and After payment the MTC Pass will be renewed.

## **CHAPTER 5**

### **CONCLUSION AND FUTURE WORK**

#### **5.1 CONCLUSION**

This project “**Digital MTC Pass using PWA**” uses the concept of Progressive Web App to make the application more user - friendly for the bus commuters. This provides the easy platform to buy and renew the MTC pass.

This reduces renewal time and improves the way of using MTC Passes. Instead of buying or using different passes for different needs , This single app takes care of all types of Bus Passes and their Renewal Process.

#### **5.2 FUTURE IDEA**

1. In future, a Single PVC Card can be created , which replicates the work of this app.
2. Will include bus pass process for Handicap
3. Will replace the mockup payment with payment gateway.

## APPENDICES

### A. SAMPLE CODE

#### 1. FARE CALCULATION

```
include 'connection.php';
session_start();
$usname = $_SESSION['Name'];
$usphone = (String)$_SESSION['Phone'];
$sql = "SELECT * FROM users WHERE Phone='$usphone' and Name = '$usname'";
$res = mysqli_query($conn, $sql);
if (mysqli_num_rows($res) > 0)
{
    while ($row = mysqli_fetch_assoc($res))
    {
        $usrid = $row['mtcid'];
        $urname = $row['Name'];
        $urphone = $row['Phone'];
        $type = $row['Type'];
        $from1 = $row['From1'];
        $from2 = $row['From2'];
        $to1 = $row['To1'];
        $to2 = $row['To2'];
        $email = $row['Email'];
    }
}

$fare = array(120.00,140.00,160.00,180.00,200.00,220.00,240.00,260.00, 280.00);
function clean($string)
{
    $string = str_replace(' ', "", $string);
    return preg_replace('/[^A-Za-z0-9\-\_]/', "", $string);
}
function stage_calc($from, $to)
{
    include 'connection.php';
    $max_stage_array = array();
    $sql = "SELECT * FROM routes";
    $res = mysqli_query($conn, $sql);
    $name = "";
    if (mysqli_num_rows($res) > 0)
    {
        while ($row = mysqli_fetch_assoc($res))
        {
```

```

$name = $row["bus_route"];
$array = explode(" ", $name);
if ((in_array($from, $array)) && (in_array($to, $array)))
{
    $start_point_stage = array_search($from, $array);
    $end_point_stage = array_search($to, $array);
    $temp = abs($start_point_stage - $end_point_stage);
    array_push($max_stage_array, $temp);
}
}
}
}
$max_stage = max($max_stage_array);
return ($max_stage + 1);
}
if ($type == 'college_one_way')
{
    $stage = stage_calc(clean($from1) , clean($to1));
}
else if ($type == 'college_two_way')
{
    $temp_stage1 = stage_calc(clean('AMB OT') , clean('THIRUMANGALAM'));
    $temp_stage2 = stage_calc(clean('PALLAVARAM') , clean('IRUMBULIYUR'));
    $stage = $temp_stage1 + $temp_stage2;
}

if ($stage-1 >= 9) {
    $temp_far = 280;
    $sql1 = "UPDATE users SET Fare = '$temp_far' WHERE Phone='$susphone' and Name = '$susname'";
    mysqli_query($conn, $sql1);
}
else
{
    $temp_far = $fare[$stage - 1];
    $sql1 = "UPDATE users SET Fare = '$temp_far' WHERE Phone='$susphone' and Name = '$susname'";
    mysqli_query($conn, $sql1);}

```

## 2. REGISTRATION

```

<?php
require_once('connection.php');
date_default_timezone_set("Asia/Calcutta");
$username=$_POST['name'];

```

```

$urphone=$_POST['phone'];
$urtype = $_POST['type'];
$urfrom=$_POST['from'];
$urto=$_POST['to'];
$urfrom2=$_POST['from2'];
$urto2=$_POST['to2'];
$email=$_POST['email'];
$wor_clg = $_POST['company'];
session_start();
$_SESSION['Name'] = $urname;
$_SESSION['Phone'] = $urphone;
$output_dir = "../propic/";/* Path for file upload */
$RandomNum = time();
$ImageName = str_replace(' ','-',strtolower($_FILES['image']['name'][0]));
$ImageType = $_FILES['image']['type'][0];
$ImageExt = substr($ImageName, strrpos($ImageName, '.'));
$ImageExt = str_replace('.',",$ImageExt);
$ImageName = preg_replace("/\.[^\s]{3,4}$/", "", $ImageName);
$NewImageName = $ImageName.'-'.$RandomNum.'-'.$ImageExt;
$ret[$NewImageName]= $output_dir.$NewImageName;
if (!file_exists($output_dir)){
    @mkdir($output_dir, 0777);
}
move_uploaded_file($_FILES["image"]["tmp_name"][0],$output_dir."/".$NewImageName );
$output_dir2 = "../aadhaar/";/* Path for file upload */
$RandomNum2 = time();
$ImageName2 = str_replace(' ','-',strtolower($_FILES['aadhaar']['name'][0]));
$ImageType2 = $_FILES['aadhaar']['type'][0];

$ImageExt2 = substr($ImageName2, strrpos($ImageName2, '.'));
$ImageExt2 = str_replace('.',",$ImageExt2);
$ImageName2 = preg_replace("/\.[^\s]{3,4}$/", "", $ImageName2);
$NewImageName2 = $ImageName2.'-'.$RandomNum2.'-'.$ImageExt2;
$ret[$NewImageName2]= $output_dir2.$NewImageName2;
if (!file_exists($output_dir2)) {
    @mkdir($output_dir2, 0777);
}
move_uploaded_file($_FILES["aadhaar"]["tmp_name"][0],$output_dir2."/".$NewImageName2
);
$output_dir3 = "../proof/";
$RandomNum3 = time();
$ImageName3 = str_replace(' ','-',strtolower($_FILES['proof']['name'][0]));
$ImageType3 = $_FILES['proof']['type'][0];
$ImageExt3 = substr($ImageName3, strrpos($ImageName3, '.'));
$ImageExt3 = str_replace('.',",$ImageExt3);
$ImageName3 = preg_replace("/\.[^\s]{3,4}$/", "", $ImageName3);

```

```

$NewImageName3 = $ImageName3.'-'. $RandomNum3.'-'. $ImageExt3;
$ret[$NewImageName3]= $output_dir3.$NewImageName3;
if (!file_exists($output_dir3)) {
    @mkdir($output_dir3, 0777);
}
move_uploaded_file($_FILES["proof"]["tmp_name"][0], $output_dir3."/".$NewImageName3 );
$usrid="tn".date("Y")."mtc_";
$sql1="select * from users where id=(SELECT max(id) from users) ";
$result=mysqli_query($conn,$sql1);
if($result!=false){
    $flag = 1;
    while($row=mysqli_fetch_array($result)){
        $tempid = $row["id"];
        $usrid.=(string) ($row["id"]+1);
    }
    $sql1="select * from users";
    $result=mysqli_query($conn,$sql1);
    if($result!=false){
        while($row=mysqli_fetch_array($result)){
            $tmpph = $row["Phone"];
            if($tmpph == $urphone){
                $flag = 0;
            }
        }
    }
    if($flag){
        if ($urtype == "college_one_way") {
            $month = date('m');
            $year = date('Y');
            $expmonth = $month+1;
            $expiry_reg = '11-'.sprintf("%02d", $expmonth).'-'. $year;
            $sql="INSERT INTO
users(mtcid,Name,Expiry,Phone,Due_Paid,Email,Coll_Comp,img,aadhaar,proof,Type,From1,To
1)
VALUES('$usrid','$surname','$expiry_reg','$urphone','Paid','$email','$wor_clg','$NewImageName'
,$NewImageName2,$NewImageName3,'college_one_way','$urfrom','$urto)";
$res=mysqli_query($conn,$sql);
}
elseif($urtype == "college_two_way")
{
    $month = date('m');
    $year = date('Y');
    $expmonth = $month+1;
    $expiry_reg = '11-'.sprintf("%02d", $expmonth).'-'. $year;
    $sql="INSERT INTO
users(mtcid,Name,Expiry,Phone,Due_Paid,Email,Coll_Comp,img,aadhaar,proof,Type,From1,To
1,From2,To2)
VALUES('$usrid','$surname','$expiry_reg','$urphone','Paid','$email','$wor_clg','$NewImageName'

```



```

,$NewImageName2',$NewImageName3','college_two_way',$urfrom',$urto',$urfrom2',$urto2')
";
$res=mysqli_query($conn,$sql);
}
$ntable="
CREATE TABLE `Susrid` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

  `Date1` varchar(100) NOT NULL,
  `Time1` varchar(500) NOT NULL,
  PRIMARY KEY (id)
)";
    if(!mysqli_query($conn,$ntable)){
        die('Sorry for inconvenience'. mysqli_error($conn));
    } else{
        header("Location:fare.php");
        exit();
    }
}
else{ echo "USer exists"; } ?>

```

### 3. QR CODE

```

var video = document.createElement("video");
var canvasElement = document.getElementById("canvas");
var canvas = canvasElement.getContext("2d");
var loadingMessage = document.getElementById("loadingMessage");
var outputContainer = document.getElementById("output");
var outputMessage = document.getElementById("outputMessage");
var outputData = document.getElementById("outputData");

function drawLine(begin, end, color) {
    canvas.beginPath();
    canvas.moveTo(begin.x, begin.y);
    canvas.lineTo(end.x, end.y);
    canvas.lineWidth = 4;
    canvas.strokeStyle = color;
    canvas.stroke();
}

navigator.mediaDevices.getUserMedia({ video: { facingMode: "environment" }
}).then(function(stream) {
    video.srcObject = stream;
    video.setAttribute("playsinline", true); // required to tell iOS safari we don't want fullscreen
    video.play();

```

```

        requestAnimationFrame(tick);
    });

function tick() {
    loadingMessage.innerHTML = "⌚ Loading video..."
    if (video.readyState === video.HAVE_ENOUGH_DATA) {
        loadingMessage.hidden = true;
        canvasElement.hidden = false;
        outputContainer.hidden = false;

        canvasElement.height = video.videoHeight;
        canvasElement.width = video.videoWidth;
        canvas.drawImage(video, 0, 0, canvasElement.width, canvasElement.height);
        Var imageData = canvas.getImageData(0, 0, canvasElement.width,
canvasElement.height);
        var code = jsQR(imageData.data, imageData.width, imageData.height, {
            inversionAttempts: "dontInvert",
        });
        if (code) {
            drawLine(code.location.topLeftCorner, code.location.topRightCorner, "#FF3B58");
            drawLine(code.location.topRightCorner, code.location.bottomRightCorner, "#FF3B58");
            drawLine(code.location.bottomRightCorner, code.location.bottomLeftCorner, "#FF3B58");
            drawLine(code.location.bottomLeftCorner, code.location.topLeftCorner, "#FF3B58");
            outputMessage.hidden = true;
            outputData.parentElement.hidden = false;
            outputData.innerHTML = code.data;
            console.log(code.data);
        } else {
            outputMessage.hidden = false;
            outputData.parentElement.hidden = true;
        }
    }
    requestAnimationFrame(tick); }

```

## 4. ROUTE DETECTION AND VALIDATION

```

require_once('connection.php');
$busno = $_GET['qrValue'];
session_start();
$usname=$_COOKIE['Name'];
$usphone=(String)$_COOKIE['Phone'];
$sql="SELECT * FROM users WHERE (Phone='$usphone' and Name = '$usname')";
$res=mysqli_query($conn,$sql);
$bus_route_no = array();
if (mysqli_num_rows($res) > 0) {

```

```

        while($row = mysqli_fetch_assoc($res))
        {
            $usrid = $row['mtcid'];
            $count = $row['Count'];
            $from = $row['From1'];
            $to = $row['To1'];
            $from2 = $row['From2'];
            $to2 = $row['To2'];
            $type = $row['Type'];
            $img = $row['img'];
        }
    }
    else {
        echo "invlaid";
    }

    function clean($string)
    {
        $string = str_replace(' ', "", $string);
        return preg_replace('/[^A-Za-z0-9\-\_]/', "", $string);
    }

    if ($type == 'college_one_way' || $type == 'college_two_way') {
        $sql2 = "SELECT * FROM routes WHERE bus_no = '$busno'";

        $res = mysqli_query($conn, $sql2);
        $name = "";
        if (mysqli_num_rows($res) > 0)
        {
            while ($row = mysqli_fetch_assoc($res))
            {
                $name = $row["bus_route"];
                $array = explode(",", $name);
                if ((in_array(clean($from), $array)) && (in_array(clean($to), $array)))
                {

                    header("Location:addcount.php");

                }
                else if( (in_array(clean($from2), $array)) && (in_array(clean($to2), $array)) ){
                    header("Location:addcount.php");
                }
            }
        }
    }
    else{
        $sql2 = "SELECT * FROM routes";
    }

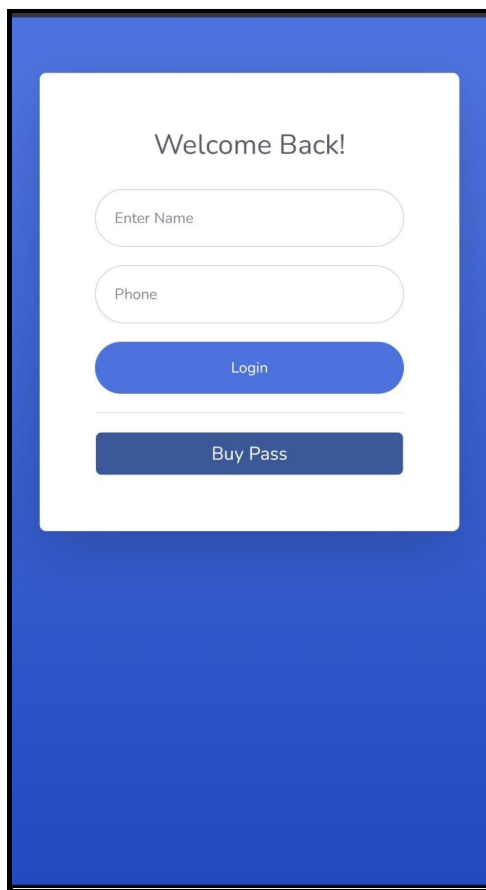
```

```

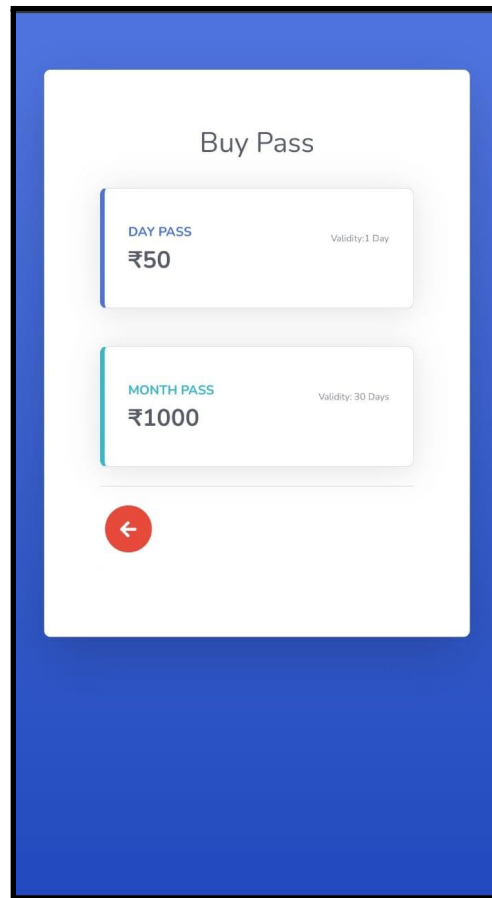
$res = mysqli_query($conn, $sql2);
while ($row = mysqli_fetch_assoc($res))
{
    array_push($bus_route_no, $row['bus_no']);
}
print_r($bus_route_no);
if (in_array($busno, $bus_route_no)){
    $_SESSION["bus_no_sess"] = $busno;
    header("Location:addcount.php");
}
}
}

```

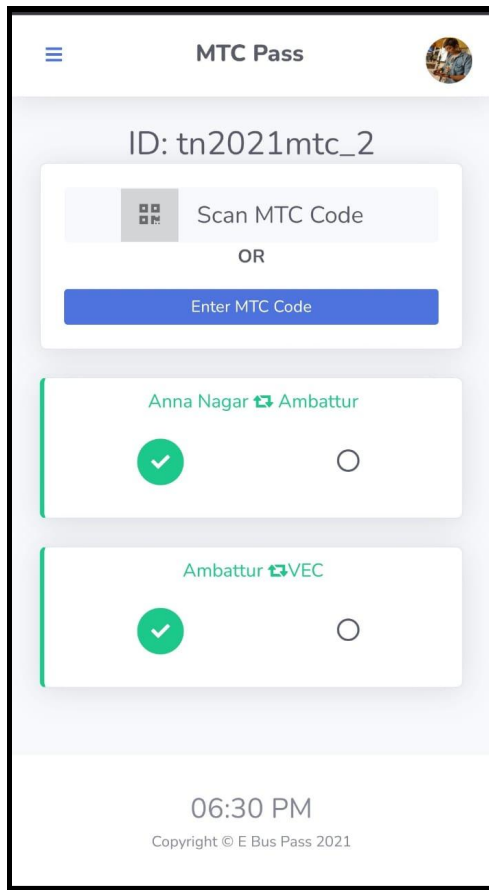
## B. SCREENSHOTS



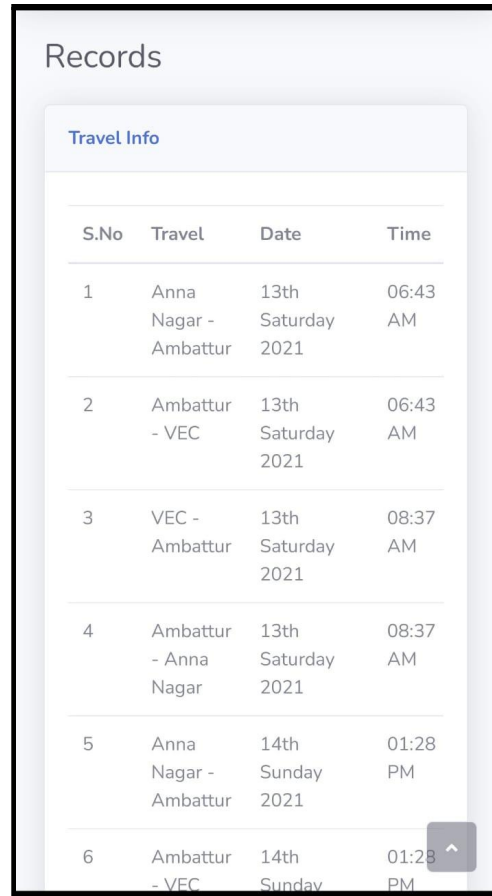
*Fig. Login Page*



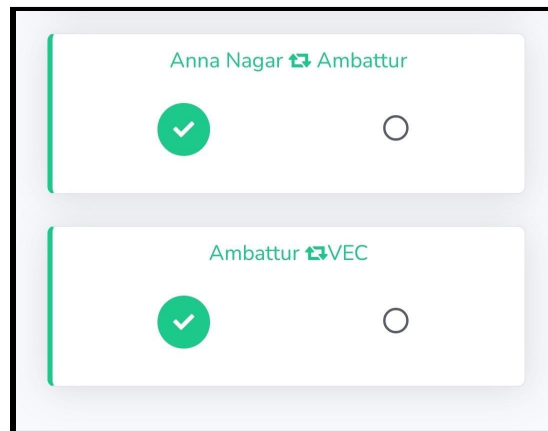
*Fig. Buy Pass*



*Fig. Dashboard*



*Fig. Trip Details*



*Fig. Trip Monitoring*

### MTC Pass Registration

**Full Name \***

**Company / College Name \***

**Email \***

**Phone number \***

**Person Type \***  
☐ College 1 Way  
☐ College 2 Way

**User picture \***

**Aadhaar Card \***

**College / Company I'd \***

---

**Route 1**  
**From \***  
  
**To \***

**Route 2**  
**From**  
  
**To**

*Fig. Registration Form*

### Payment Gateway

**Fare: 280**

To: MTC Bus Dept(Renewal)

Name: Praveen

Phone: 8825511238

**Card Owner**

**Card number**

**Expiration Date**

**CVV ?**

*Fig. Mock up Payment*

## REFERENCES

[1] Miss. Mohini S. Shirsath ,Pooja M. Chinchole,Vaishnavi R. Mahajan, Varsha G. Mogal , “A Review on Smart Bus Ticketing System using QR-Code” , International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 03 | Mar-2018

[2] Nwakanma Ifeanyi Cosmas, Etus C, Ajere I.U. & Agomuo Uchechukwu Godswill , “Online Bus Ticket Reservation System” , International Journal Of Computer Science And Statistics Vol. 1 No.2, 2015

[3] V, Pandimurugan & R, Jayaprakash & V, Rajashekar & K, Yogeshwar, “Smart Buspass System Using Android” in Conference on Innovations in Information and Communication Technology (ICIICT), Chennai, India, 2019

[4] Prof. B. A. Khivsara, Mr. Kakshil J. Jain, Mr. Nilesh M. Shinde, Miss. Kavita B.Pachorkar, Miss. Prachi A. Gandhi, “Smart Bus Ticketing System”, International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 01 | Jan-2020

## TECHNICAL PROJECT OUTCOMES

After successful completion of project work student will be able to:

TPO1: Analyze, Design and Implement projects with a comprehensive, systematic and ethical approach.

TPO2: Apply modern tools to execute and integrate modules in the project.

TPO3: Apply techniques for societal, health care, and real time sustainable research projects.

TPO4: Develop communication skills by the technical presentation activities.

TPO5: Contribute as a team and lead the team in managing technical projects.

	TPO1	TPO2	TPO3	TPO4	TPO5
DIGITAL MTC PASS USING PWA	3	3	3	3	3

**"DIGITAL MTC PASS USING PWA".**

**(Indicate as 1 - Less than 30%; 2- 30.1 -60%; 3- Above 60.1%)**