A Mini Project Report on

**Bus Ticketing System**

Submitted in partial fulfillment of the requirements for the

award of the degree of

**Bachelor of Engineering**

In

**Computer Engineering**

By

**Ashish Yadav (M221430)**

**Atreya Ganguly (M221405)**

**Harshit Mishra (M221413)**

**Mitesh Munankar (M221414)**

Under the Guidance of

**Dhananjay Raut**

****

**Department of Computer Engineering**

**Watumull Institute of Electronics Engineering and Computer Technology**

**Ulhasnagar**

UNIVERSITY OF MUMBAI

**Academic Year 2021-2022**

**Approval Sheet**

This Mini Project Report entitled “***Bus Ticketing System***” Submitted by ***"Ashish Yadav", (M221430)***, ***"Atreya Ganguly", (M221405)***, ***"Harshit Mishra", (M221413)***, ***"Mitesh Munankar", (M221414)*** is approved for the partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in ***Computer Science*** from University of Mumbai.

(Dhananjay Raut)

Guide

Head Department of Computer Engineering

Place: Ulhasnagar

Date: 2nd May, 2022

**CERTIFICATE**

This is to certify that the mini project entitled “***Bus Ticketing***” submitted by ***"Ashish Yadav", (M221430), "Atreya Ganguly", (M221405), "Harshit Mishra", (M221413), "Mitesh Munankar", (M221414)*** for the partial fulfillment of the requirement for award of a degree Bachelor of Engineering in Computer Engineering,to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

Guide Name & Signature Examiners:

1.

2.

Head Department of Computer Engineering Principal

Place: Ulhasnagar

Date: 2nd May, 2022

**Declaration**

We declare that this written submission represents our ideas in our own words and where others’ ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

———————————————

(Signature)

———————————————

(Ashish Yadav and M221430)

(Atreya Ganguly and M221405)

(Harshit Mishra and M221413)

(Mitesh Munankar and M221414)

Date: 2nd May, 2022

**Abstract**

The **Bus Ticketing System Project in JavaScript** was developed using java programming language, in this **Bus Ticketing System Project In JavaScript** pdf helps to calculate the ticket fare of the bus ticket while you wish to travel a long distance.

A online **Bus Ticketing System Project In JavaScript**will calculate the tax amount and road amount and total passenger number and gives you the detail about the ticket fare. Also, you can select the type of ticket you want to get like the Premium Class or Economy class. Also, you can select the ticket mode that is whether you want to return with the same bus or not.

**Contents**

1. **Introduction 7**
2. Problem Definition
3. Objectives
4. Scope
5. **Existing System/ Project 7**
6. **Technology Stack 8** 
   1. Code Editor
   2. Languages
7. **Benefits and Applications 15**
8. **Project Design 16**
9. Proposed System
10. Flow of Modules
11. Data Flow Diagram
12. **Project Implementation 20**
13. **Result 21**
14. **Bibliography 22**

**1.Introduction**

1.1 Problem Definition

To make bus ticketing services available online for 24/7 hours so as to easy cost and with less human interaction. but also, to make technological advancement and make users less dependent on human resources like counter attendant.

1.2 Objectives

Our objective is to provide users free simple website/software to generate e-ticket easily and conveniently for the buses of long-distance travelling

1.3 Scope

This project can be accessed anywhere over the internet using the link which is also a source code link

<https://github.com/unixm9/Mini_Project/tree/main/Bus%20Ticketing%20System>

**2.Existing System/Project**

Already many bus ticketing systems exist but majority of them are paid and complicated to use for a naïve user but our system is more user friendly than other counter parts.

**3.Technology Stack**

3.1 Code Editor

**Microsoft Visual Studio Code**

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

**Language support**

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

**Data collection**

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. In addition, because of the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected. According to Microsoft, the data is shared with Microsoft-controlled affiliates and subsidiaries, although law enforcement may request it as part of a legal process.

**Version control**

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature, you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as make push and pull requests directly from the Visual Studio Code program.

3.2 Languages

**HTML**

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 (CSS) 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 <img /> and <input /> directly introduce content into the page. Other tags such as <p> 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 (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**HTML Basic Markup**

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic "Hello, World!" program:

<!DOCTYPE html>

<html>

<head>

<title>This is a title</title>

</head>

<body>

<div>

<p>Hello world!</p>

</div>

</body>

</html>

The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text <title>This is a title</title> defines the browser page title shown on browser tabs and window titles, and the tag <div> defines a division of the page used for easy styling.

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

**CSS**

Cascading Style Sheets (CSS) 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; and enable 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 feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), 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 (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). 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.

**Basic CSS Code**

<style>

h1 {

color: blue;

}

</style>

**JavaScript**

JavaScript (/ˈdʒɑːvəˌskrɪpt/),[10] often abbreviated as JS, is a programming language that conforms to the ECMAScript specification.[11] JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype-based object-orientation and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

**Run-time environment**

JavaScript typically relies on a run-time environment (e.g., a web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a web page DOM). These environments are single-threaded. JavaScript also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML <script> elements). This is not a language feature per se, but it is common in most JavaScript implementations. JavaScript processes messages from a queue one at a time. JavaScript calls a function associated with each new message, creating a call stack frame with the function's arguments and local variables. The call stack shrinks and grows based on the function's needs. When the call stack is empty upon function completion, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output is performed using events and callback functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information.

**4.Benefits and Applications**

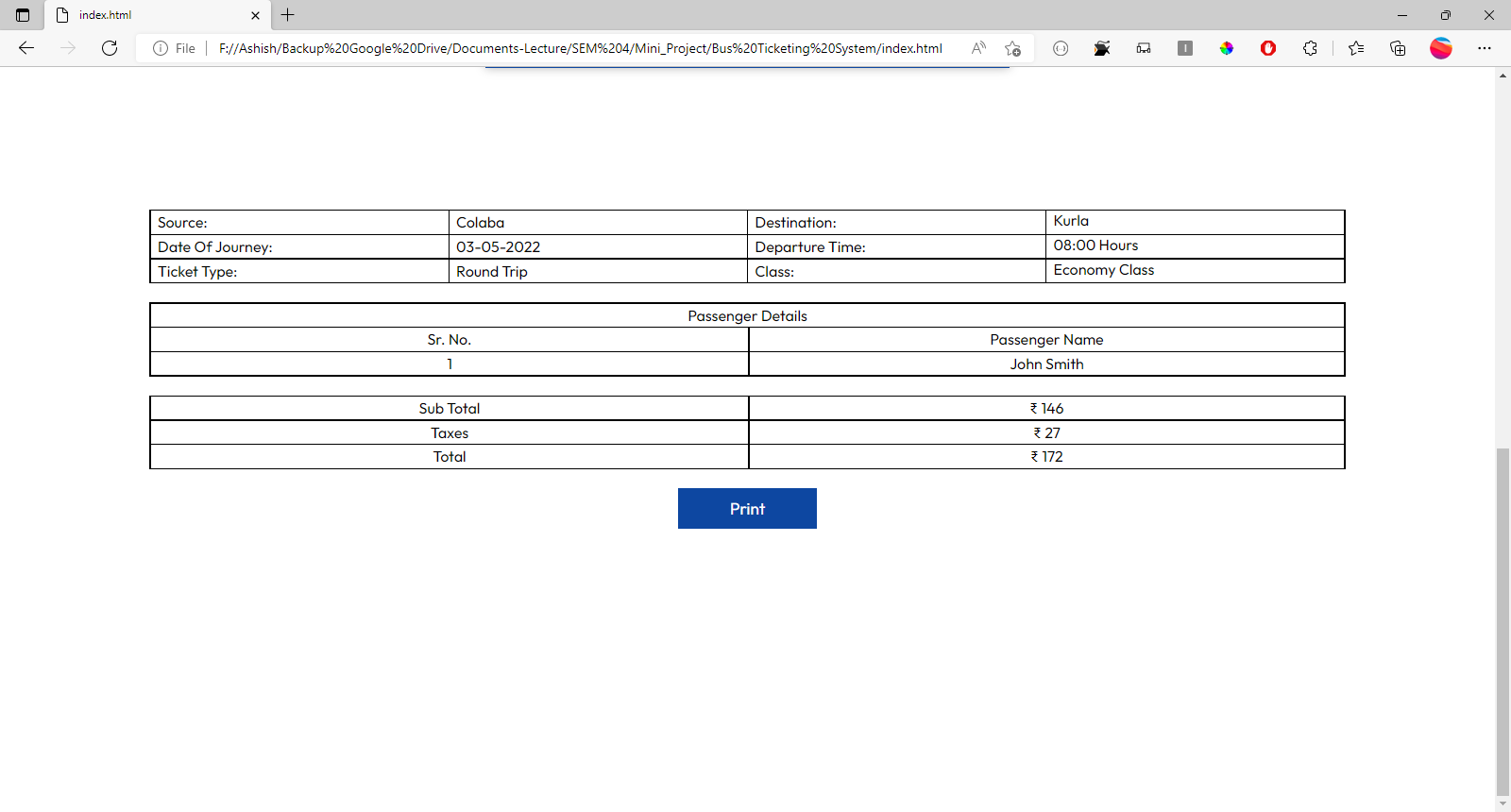
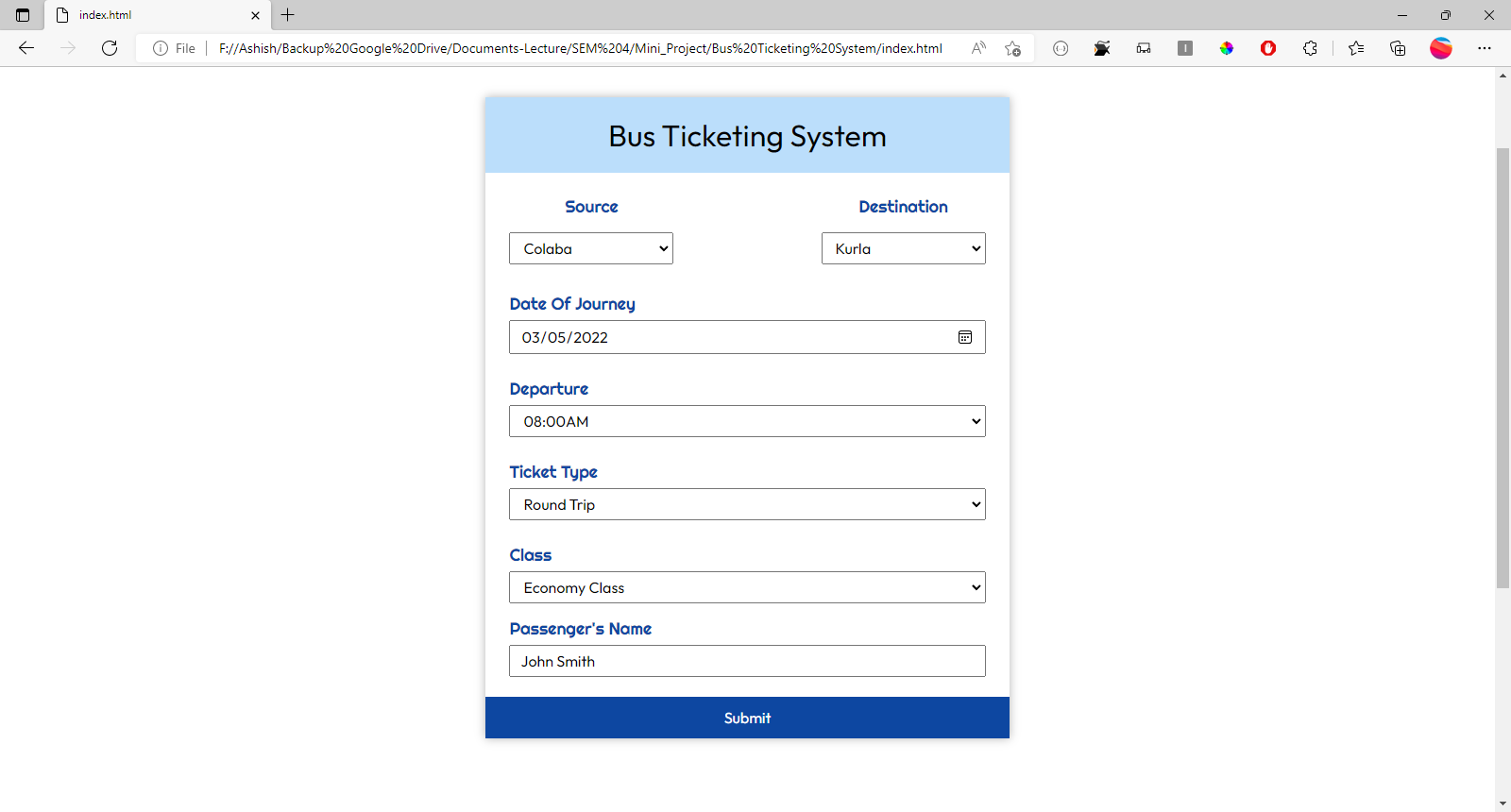
1)Simple and Easy to use: The system is simple and easy to use. All the options are provided in the software for smooth functioning.

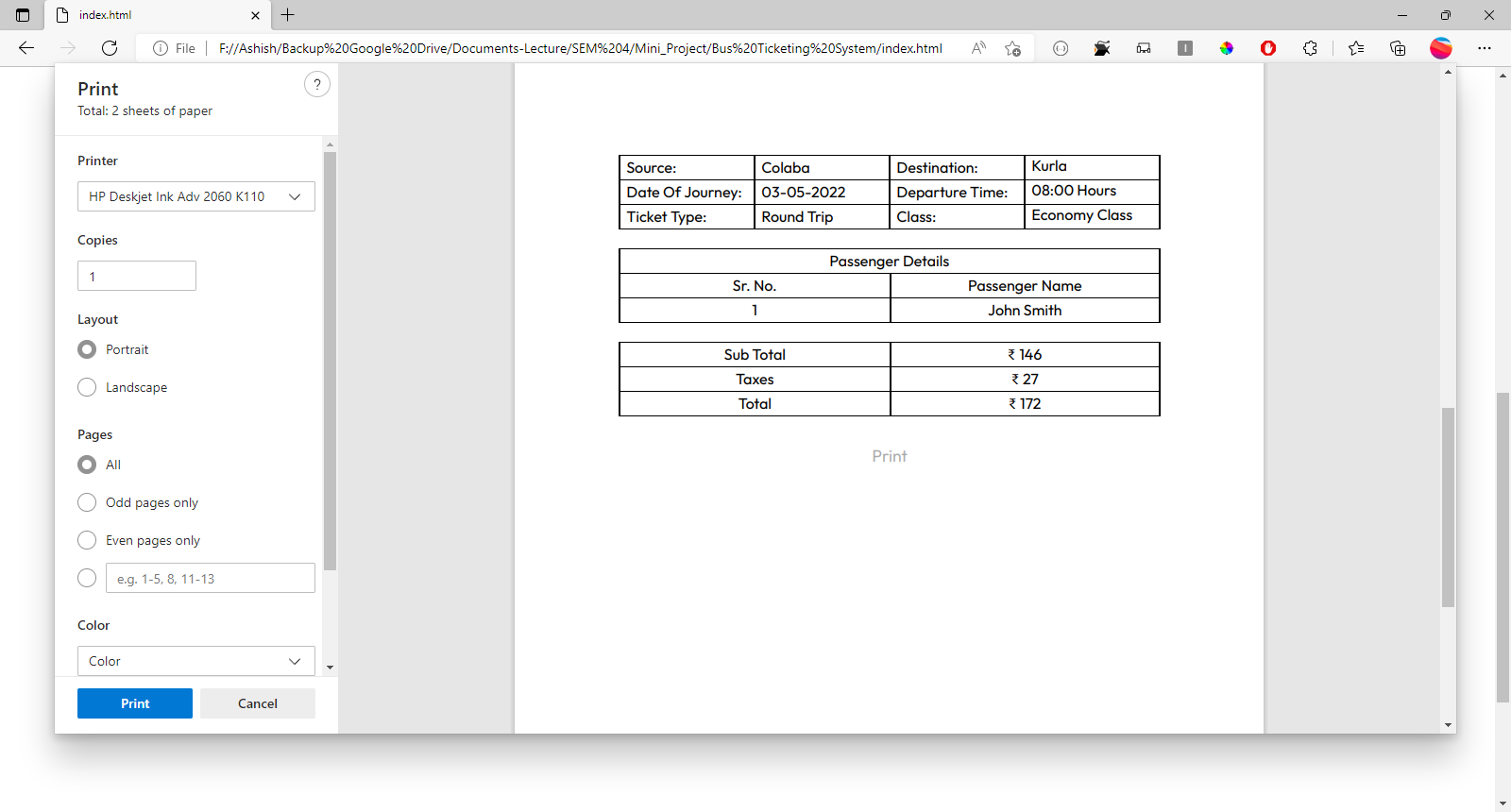
2)Highly secure: The system is highly secure and reliable to use in self ticket booking system at bus stations.

3)Innovative: The system is based on new technology.

**5.Project Design**

5.1Proposed System

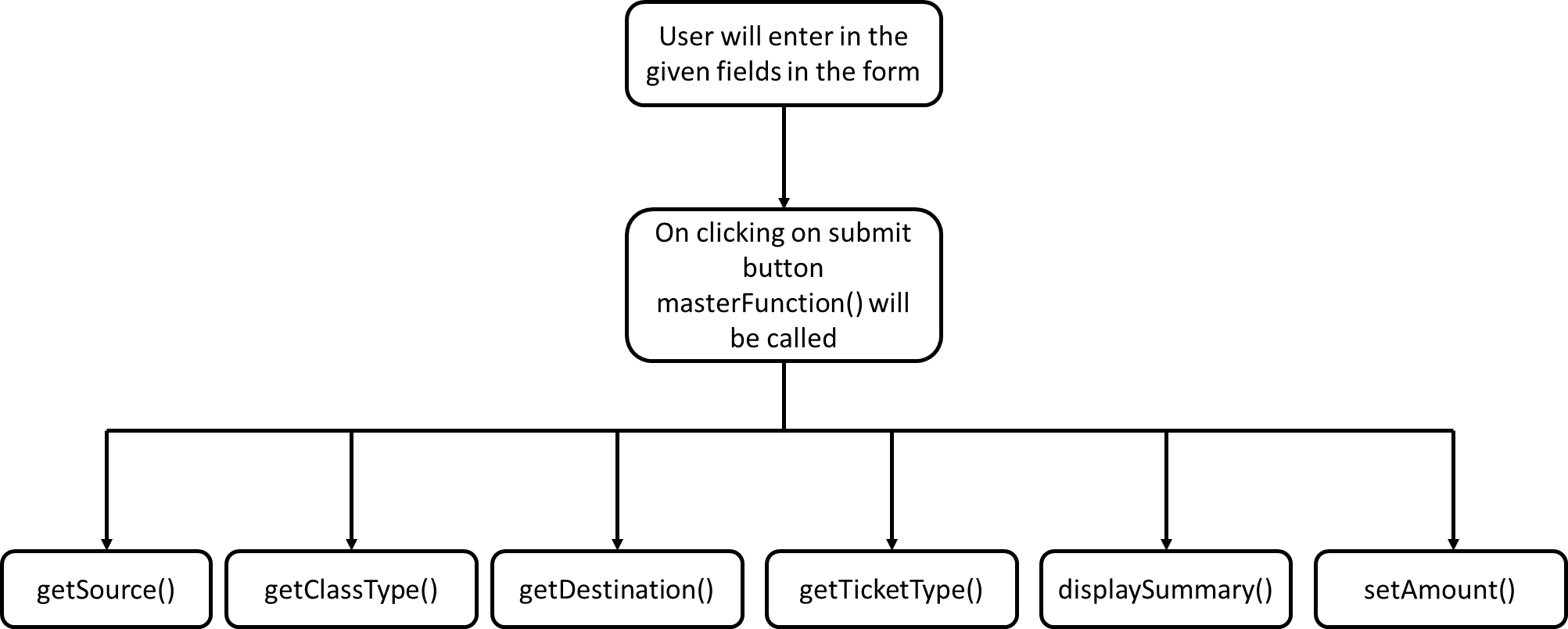




5.2 Flow of Modules

1. First User will Select Source and destination using drop down menu
2. Then Date of journey can be chosen using calendar dialog box by the user
3. User will select departure time using dropdown menu of predefined timings
4. Type of class and ticket type can be chosen using respective menus
5. User’s name will be entered in the last text box
6. After entering all the necessary details submit will be clicked by the user and printable ticket will be generated

5.3 Data Flow Diagram



**6.Project Implementation**

Source Code

<https://github.com/unixm9/Mini_Project/tree/main/Bus%20Ticketing%20System>

Our project is available on the internet and can be accessed anywhere in the world. The source code can also be publicly viewed by the users.

**7.Result**

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements.

**8.Bibliography**

1. suarez. “Bus Reservation System Project, *Itsourcecode.Com*, 4 Mar. 2021, <https://itsourcecode.com/free-projects/java-projects/bus-reservation-system-project-in-java-with-source-code/>
2. “W3Schools Online Web Tutorials.” *W3Schools Online Web Tutorials*, <https://www.w3schools.com/>
3. “MDN Web Docs.” *MDN Web Docs*, <https://developer.mozilla.org/en-US/>

HOD: Dhananjay Raut

College: Watumull Institute of Electronics Engineering and Computer Technology,Ulhasnagar

**Acknowledgement**

We have great pleasure in presenting the mini project report on **Bus Ticketing System**. We take this opportunity to express our sincere thanks towards our guide **Professor Dhananjay Raut** Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof.Sachin Malave** Head of Department, Computer Engineering, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Student Name1: Ashish Yadav**

**Student ID1: (M221430)**

**Student Name2: Atreya Ganguly**

**Student ID2: (M221405)**

**Student Name3: Harshit Mishra**

**Student ID3: (M221413)**

**Student Name4: Mitesh Munankar**

**Student ID4: (M221414)**