

**PROJECT DOCUMENTATION**  
**Bus Details Management Project**  
**BUSSIN.com**



By

**ALI MOHAMMED S  
ASHTAMI A S  
FAHAD SAJEEM  
FEBA JOSE  
MUHAMMED SAFAR  
SREEDEV T S**

# **CONTENTS**

1. Introduction	2
2. Objectives	3
3. Features	4
4. System Design	6
I. Software Requirements	6
II. UI Design (Homepage)	6
III. Features Design and Implementation	7
IV. Backend Design	11
5. Result	15

# **Introduction**

A wide range of people in our community use public transport these days. It is a requirement for them to know the availability of the buses, its schedule and more details. The Bus Details Management System is a comprehensive solution designed to simplify the management of bus details, from inputting and viewing to filtering and exporting data. This system provides a user-friendly interface for users to interact with, allowing them to seamlessly perform various operations related to bus management.

# **Objectives**

The objectives of the Bus Details Management project are as follows:

## **1. Efficient Bus Details Management**

The primary objective is to develop a system that allows for efficient management of bus details, including inputting, storing, and retrieving information about various buses.

## **2. User-friendly Interface**

Develop a user-friendly interface that enables users to easily input bus details through a form and view them in a structured manner in a table format.

## **3. Data Export**

Enable users to export the entered bus details in a JSON format, allowing for easy sharing and external data analysis.

## **4. Filtering and Search Functionality**

Implement functionality that allows users to filter and search for available buses based on different criteria such as source, destination and time range.

## **5. Seat Booking Management**

Allow users to book or leave seats on available buses directly through the system, streamlining the seat booking process.

## **6. Provide functionality to view counts of buses based on their type.**

# Features

## 1. Input Bus Details

A detailed information about buses, including unique bus name, bus type, source, destination, departure time, and available seats, can be input through a user-friendly form.

## 2. View Bus Details

The system displays the entered bus details in a structured table format below the input form, providing users with an overview of all available buses at a glance.

## 3. Export Bus Details

It has a functionality to export the entered bus details as a JSON file by clicking on the export button.

## 4. Filter Available Buses

The system allows users to filter available buses based on various criteria, including:

- Source
- Destination
- Both source and destination
- Time range

These filtering options enable users to quickly find buses that match their specific requirements.

## **5. Bus Type Counts**

Users can view counts of buses based on their type, providing insights into the distribution of different bus types within the system.

## **6. Seat Booking**

The system facilitates seat booking operations, allowing users to reserve or release seats on available buses directly through the interface.

# **System Design**

## **I. Software Requirements**

### **Front-end**

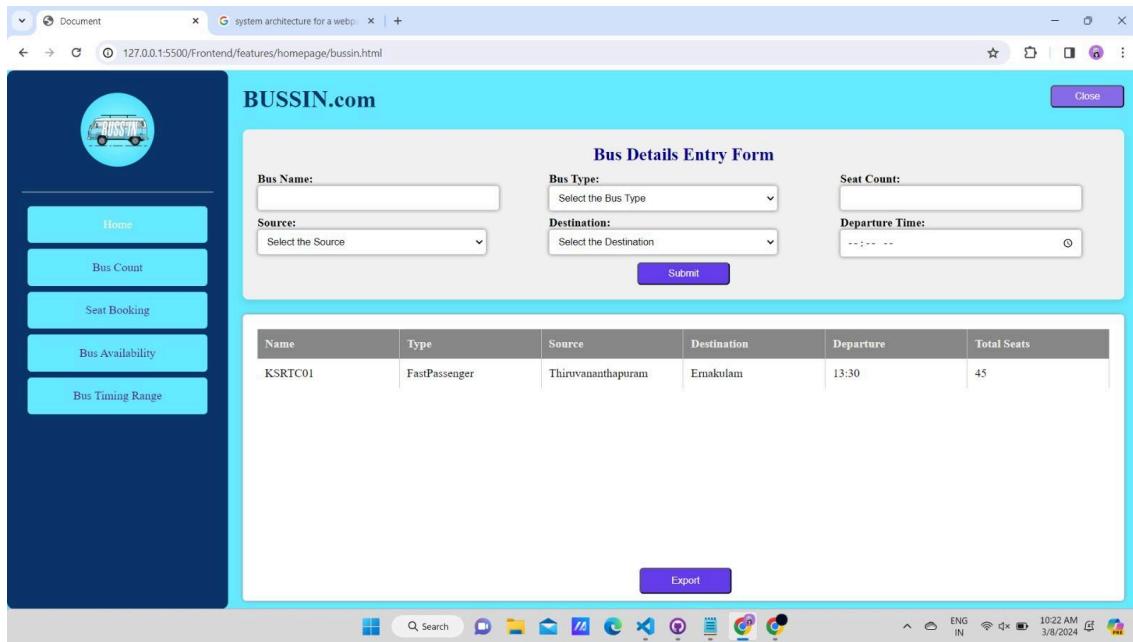
HTML, CSS and JavaScript are used for rendering the UI. This includes structuring the webpages, data validation, styling each component, collecting information from the user and displaying filtered data received from the backend in the webpage set for each activity.

### **Back-end**

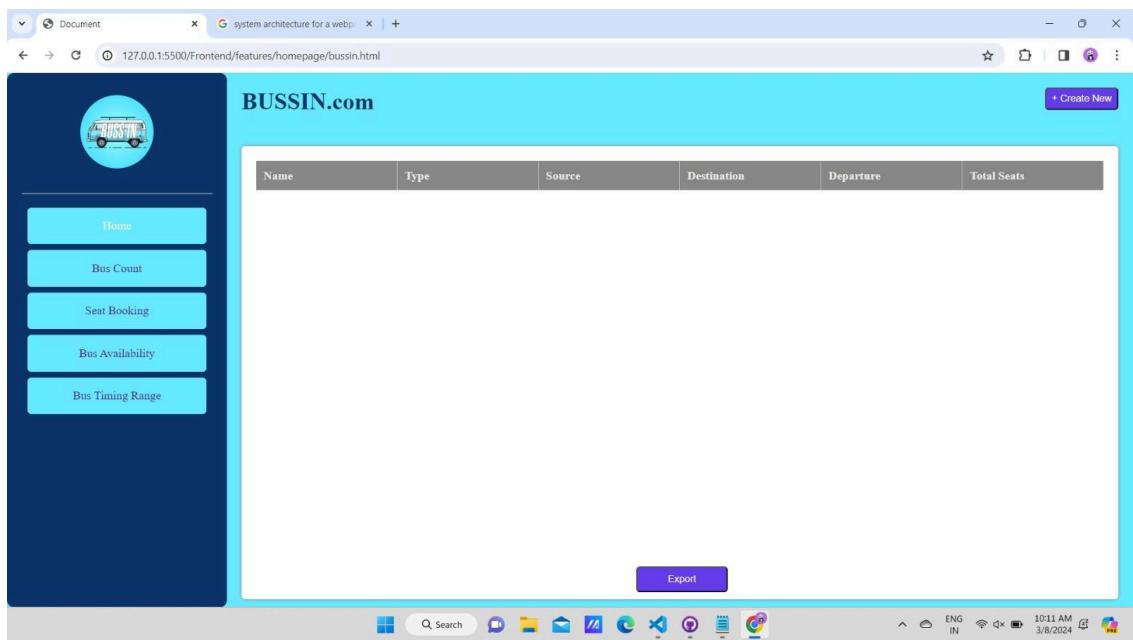
Java is used for filtering the data based on the requirements. The filtering activities performed using Java are bus type filtering, time range filtering, source-destination filtering. It also updates the seat count of the buses based on the booking details in the backend

## **II. UI Design (Homepage)**

The homepage has a sidebar for navigation with the logo on its top. It has a create new button on the top right corner. It opens a form when clicked in which we can fill the bus details which includes bus name, bus type, seat count, source, destination and departure time. Data validation check is implemented on each field. When the submit button is clicked after inputting all the details of a bus, it is displayed in the table below it. At the bottom of the table, there is an export button. When the export button is clicked, the data is downloaded as a json file but if there's no data in the table, it shows an alert box that there is no data to export.



Home page with bus details entry form



Default homepage

### III. Features Design and Implementation

#### Bus Count

This page displays all the types of buses with their total counts in a page.

The screenshot shows a web browser window with the title "Document" and URL "127.0.0.1:5500/Frontend/features/buscount/busCount.html". The page has a dark blue sidebar on the left with a logo and five buttons: "Home", "Bus Count", "Seat Booking", "Bus Availability", and "Bus Timing Range". The main content area is titled "BUSSIN.com" and contains a table with two columns: "Bus Type" and "Count". The data is as follows:

Bus Type	Count
Fast Passenger	6
SuperFast	8
Ordinary	5

At the bottom right of the screen, there is a system tray with icons for battery, signal, and date/time (10:12 AM, 3/8/2024).

Bus Count Page

## Seat Booking

The screenshot shows a web browser window with the title "Document" and URL "127.0.0.1:5500/Frontend/features/seatbooking/seatBooking.html". The sidebar and layout are identical to the Bus Count page. The main content area is titled "BUSSIN.com" and contains a "Book Your Tickets!" section with a dropdown menu for "Bus Name" (set to "Select the Bus") and radio buttons for "Reservation" and "Leave". Below this is a "Submit" button. Further down is a table with columns "Activity" and "BusName". The data is as follows:

Activity	BusName
Reservation	KSRTC01
Leave	KSRTC01

At the bottom right of the screen, there is a system tray with icons for battery, signal, and date/time (10:25 AM, 3/8/2024). A "Export" button is visible at the bottom of the table.

Seat Booking Page

This feature allows the user to reserve or leave a seat of a particular bus with its unique name. It incorporates a table below which shows the bookings and it can be exported as a json file. This file can be used to check the condition where there may be chances of conflicts regarding the count of seats and the booking.

## Bus Availability

The bus availability page enables a user to check the details of all the buses either from a particular source or a particular destination or both. If the user wants to see only the details of buses from a source, he can select a place from the options of source and on clicking the search button, the details are displayed on the table below. Similarly, if the user wants to see only the details of buses from a source, he can select a place from the options of source and on clicking the search button, the details are displayed on the table below. The same working is implemented for the case where both the source and destination are given.

The screenshot shows a web browser window with the URL [127.0.0.1:5500/Frontend/features/sourcedestinationfilter/sourceDest.html](http://127.0.0.1:5500/Frontend/features/sourcedestinationfilter/sourceDest.html). The page title is "Find Your Bus!" and it displays a table of bus details. The left sidebar has a navigation menu with "Home", "Bus Count", "Seat Booking", "Bus Availability" (which is highlighted in blue), and "Bus Timing Range".

Name	Type	Source	Destination	Departure	Total Seats
KSRTC01	SuperFast	Thiruvananthapuram	Kollam	14:25	30
KSRTC08	Ordinary	Thiruvananthapuram	Kottayam	10:40	35
KSRTC10	FastPassenger	Thiruvananthapuram	Kollam	20:50	50
KSRTC13	Ordinary	Thiruvananthapuram	Kollam	22:05	45
KSRTC14	SuperFast	Thiruvananthapuram	Pathanamthitta	16:30	55

List of details of buses from Thiruvananthapuram

The screenshot shows a web browser window with the URL [127.0.0.1:5500/Frontend/features/sourcedestinationfilter/sourceDest.html](http://127.0.0.1:5500/Frontend/features/sourcedestinationfilter/sourceDest.html). The page title is "BUSSIN.com". On the left, there is a sidebar with a bus icon and five buttons: Home, Bus Count, Seat Booking, Bus Availability, and Bus Timing Range. The main content area has a header "Find Your Bus!" with dropdown menus for "Source" (set to "Select the Source") and "Destination" (set to "Kollam"). A "Search" button is located below the dropdowns. Below the search bar is a table listing bus details:

Name	Type	Source	Destination	Departure	Total Seats
KSRTC01	SuperFast	Thiruvananthapuram	Kollam	14:25	30
KSRTC10	FastPassenger	Thiruvananthapuram	Kollam	20:50	50
KSRTC12	SuperFast	Wayanad	Kollam	02:43	52
KSRTC13	Ordinary	Thiruvananthapuram	Kollam	22:05	45
KSRTC19	FastPassenger	Malappuram	Kollam	08:55	52

List of details of buses heading to Kollam

The screenshot shows a web browser window with the same URL and layout as the previous one. The "Source" dropdown is now set to "Thiruvananthapuram" and the "Destination" dropdown is set to "Kollam". The table below shows the results:

Name	Type	Source	Destination	Departure	Total Seats
KSRTC01	SuperFast	Thiruvananthapuram	Kollam	14:25	30
KSRTC10	FastPassenger	Thiruvananthapuram	Kollam	20:50	50
KSRTC13	Ordinary	Thiruvananthapuram	Kollam	22:05	45

List of buses from Thiruvananthapuram to Kollam

## Bus Timing Range

This page enables the user to find all the buses at a particular time period by selecting a range from the options given.

The screenshot shows a web browser window with the title 'Document' and the URL '127.0.0.1:5500/Frontend/features/bustiming/busTiming.html'. The page has a dark blue sidebar on the left with a bus icon and five menu items: Home, Bus Count, Seat Booking, Bus Availability, and Bus Timing Range. The main content area has a light blue header with the text 'BUSSIN.com' and 'Get the Details you Wish!'. Below this is a form with a radio button group for 'Select the time range' with options: 8 am to 12 pm, 12 pm to 4 pm, 4 pm to 8 pm, and 8 pm to 8am, with the last option selected. A 'Submit' button is next to it. Below the form is a table with the following data:

Name	Type	Source	Destination	Departure	Total Seats
KSRTC02	FastPassenger	Kollam	Pathanamthitta	20:30	50
KSRTC03	Ordinary	Ernakulam	Kottayam	02:45	35
KSRTC04	SuperFast	Kozhikode	Pathanamthitta	23:00	50
KSRTC06	FastPassenger	Kozhikode	Kannur	07:35	50
KSRTC09	FastPassenger	Kollam	Kasaragod	20:00	35
KSRTC10	FastPassenger	Thiruvananthapuram	Kollam	20:50	50
KSRTC11	Ordinary	Thrissur	Ernakulam	04:42	39
KSRTC12	SuperFast	Wayanad	Kollam	02:43	52
KSRTC13	Ordinary	Thiruvananthapuram	Kollam	22:05	45
KSRTC17	FastPassenger	Pathanamthitta	Palakkad	01:00	30

List of buses from 8pm to 8am

## IV. Backend Design

The csvReader class converts the csv files into an arraylist for data processing, updating and filtering. Bus class stores all the bus objects and the reserve class stores all the reservation objects. Class reservation updates the seat count based on reserve data. Time class filters the data based on time range. TypeFilter class filters the data based on the types of bus and provides its count. SourceDestination class filters the data based on the source and destination as per user's requirement. CsvWriter class converts the arraylist into a csv file.

```

"C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe" "-javaagent:C:\Users\Asus\AppData\Local\Temp\jdt-launcher-20230720-1133.jar" -Dfile.encoding=UTF-8
FILE READ SUCCESSFULLY FORM RESERVATION FILE
FILE READ SUCCESSFULLY FORM BUS FILE
=====
SEAT UPDATER IS RUNNING

SEAT RESERVED FOR KSRTC05
NO SEATS ARE RESERVED AFTER CANCEL 0 FOR KSRTC07
SEAT RESERVED FOR KSRTC03
NO SEATS ARE RESERVED AFTER CANCEL 0 FOR KSRTC04
=====

1.TypeFiltering
2.SourceDestinationFiltering
3.TimeFiltering
4.Reservation
5.Exit
Enter the option:1
=====
=====

1.TypeFiltering
2.SourceDestinationFiltering
3.TimeFiltering
4.Reservation
5.Exit
Enter the option:2

Enter the option:2
=====
SourceDestinationFiltering
1.SourceFilter
2.DestinationFiltering
3.SourceDestinationFiltering
4.Exit
Enter the option:1
Source Filter....
Enter the Source:Thiruvananthapuram
SourceDestinationFiltering
1.SourceFilter
2.DestinationFiltering
3.SourceDestinationFiltering
4.Exit
Enter the option:2
Destination Filter....
Enter the Destination:Kollam
SourceDestinationFiltering
1.SourceFilter
2.DestinationFiltering
3.SourceDestinationFiltering
4.Exit
Enter the option:3
Source Destination Filter....
Enter the Source:Thiruvananthapuram
Enter the Destination:Kollam

```

```
Enter the Destination:KULLAM
SourceDestinationFiltering
1.SourceFilter
2.DestinationFiltering
3.SourceDestinationFiltering
4.Exit
Enter the option:4
=====
1.TypeFiltering
2.SourceDestinationFiltering
3.TimeFiltering
4.Reservation
5.Exit
Enter the option:3
=====
TimeFiltering
1.08:00 to 12:00
2.12:00 to 16:00
3.16:00 to 20:00
4.20:00 to 08:00
5.Exit
Enter the option:1
Morning Filter....
FILE SCHEDULE CREATED SUCCESSFULLY
TimeFiltering
1.08:00 to 12:00
2.12:00 to 16:00
3.16:00 to 20:00
```

```
FILE SCHEDULE CREATED SUCCESSFULLY
TimeFiltering
1.08:00 to 12:00
2.12:00 to 16:00
3.16:00 to 20:00
4.20:00 to 08:00
5.Exit
Enter the option:2
AfterNoon Filter....
FILE SCHEDULE CREATED SUCCESSFULLY
TimeFiltering
1.08:00 to 12:00
2.12:00 to 16:00
3.16:00 to 20:00
4.20:00 to 08:00
5.Exit
Enter the option:3
Evening Filter Filter....
FILE SCHEDULE CREATED SUCCESSFULLY
TimeFiltering
1.08:00 to 12:00
2.12:00 to 16:00
3.16:00 to 20:00
4.20:00 to 08:00
5.Exit
Enter the option:4
Night Filter....
TimeFiltering
```

```
1.TypeFiltering
2.SourceDestinationFiltering
3.TimeFiltering
4.Reservation
5.Exit
Enter the option:5
=====
```

```
Process finished with exit code 0
```

User console

## **Result**

The website frontend was tested for its working, and the tests were successful and output was verified. This data was then converted from JSON to CSV ,and sent to the backend team. The backend team then tested this data to check functionality of the backend program. The tests were successful and the data was filtered based on the necessary requirements. This data was then sent to the front end team after converting from CSV to JSON .This data was then displayed on the website when required. Overall all tests on the current data was successful.