

A screenshot of the Visual Studio Code (VS Code) interface. The window title is "File Edit Selection View Go Run ...". The top bar includes a search field "c++coding" and various icons for file operations like copy, paste, and close.

The left sidebar contains the "EXPLORER" view, which shows a tree structure of files and folders under the "C++CODING" category. The files listed include ".vscode", "project\_mini.cpp", "mini\_project.cpp", "mini\_project.exe", "mini\_project2.cpp", "settings.json", "tasks.json", "practice.cpp", "inheritance.cpp", "areaofcircle.cpp", "areaofcircle.exe", "class.cpp", "class.exe", "classAndStruct.CPP", "concatenatesttwosrings.cpp", "concatenatesttwosrings.exe", "conditional.cpp", "conditional.exe", "controal\_statement.cpp", "controal\_statementexe", "experiment1.cpp", "experiment1.exe", "firstc++.cpp", "firstc++.exe", and "OUTLINE" and "TIMELINE".

The main editor area displays the content of "mini\_project2.cpp". The code is as follows:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 /*
5 Simple Bus/Train Booking System - Console based
6 Features:
7 - View routes
8 - Admin: Add/Delete routes
9 - Book ticket (auto ticket id generation)
10 - Cancel ticket
11 - View my bookings (by phone or ticket id)
12 - Data persistence using text files: routes.txt, bookings.txt
13
14 Files format (simple):
15 routes.txt line format:
16 route_id|type|source|destination|date|time|fare|total_seats|booked_seats
17
18 bookings.txt line format:
19 ticket_id|route_id|passenger_name|phone|seats_booked|seat_numbers(comma)|total_fare
20 */
21
22 struct Route {
23     int route_id;
24     string type; // "Bus" or "Train"
25     string source, destination;
26     string date; // simple dd-mm-yyyy
27     string time; // hh:mm
28     double fare;
29     int total_seats;
30     int booked_seats;
```

The status bar at the bottom shows "Ln 218, Col 20" and other settings like "Spaces: 4", "UTF-8", "CRLF", "C++", "Win32", and "Prettier".

The screenshot shows the Visual Studio Code (VS Code) interface. The title bar includes the file path ".vscode > project\_mini.cpp > C++ mini\_project2.cpp". The left sidebar has icons for Explorer, Search, Find, Go To, Task, and Outline. The Explorer view shows a tree with a folder named "C++CODING" containing files like ".vscode", "project\_mini.cpp", "mini\_project.cpp", "mini\_project.exe", "mini\_project2.cpp", "settings.json", "tasks.json", "practice.cpp", and "inheritance.cpp". The "inheritance.cpp" file is currently selected. The main editor area displays the following C++ code:

```
22 struct Route {
23     int booked_seats;
24 };
25
26 struct Booking {
27     long long ticket_id;
28     int route_id;
29     string passenger_name;
30     string phone;
31     int seats_booked;
32     string seat_numbers; // for simplicity store as "1,2,3"
33     double total_fare;
34 };
35
36 vector<Route> routes;
37 vector<Booking> bookings;
38
39 // Utility helpers
40 vector<string> split(const string &s, char delim) {
41     vector<string> out;
42     string cur;
43     stringstream ss(s);
44     while (getline(ss, cur, delim)) out.push_back(cur);
45     return out;
46 }
47
48 void loadRoutes(const string &filename="routes.txt") {
49     routes.clear();
50     ifstream fin(filename);
51     if(!fin) return;
52 }
```

The status bar at the bottom shows "Ln 218, Col 20" and other settings like "Spaces: 4", "UTF-8", "CRLF", "C++", "Win32", and "Prettier".

A screenshot of the Visual Studio Code (VS Code) interface, version 1.68.2, showing a C++ project named "mini\_project2".

The Explorer sidebar on the left shows the project structure:

- .vscode
- project\_mini.cpp
- mini\_project.cpp
- mini\_project.exe
- mini\_project2.cpp
- settings.json
- tasks.json
- practice.cpp
- inheritance.cpp (selected)
- areaofcircle.cpp
- areaofcircle.exe
- class.cpp
- class.exe
- classAndStruct.CPP
- concatenatesttwosringss.cpp
- concatenatesttwosringse.exe
- conditional.cpp
- conditional.exe
- controal\_statement.cpp
- controal\_statement.exe
- experiment1.cpp
- experiment1.exe
- firstc++.cpp
- firstc++.exe

The "mini\_project2.cpp" editor tab is open, displaying the following code:

```
55 void loadRoutes(const string &filename="routes.txt") {  
56     ifstream fin(filename);  
57     if(!fin) return;  
58     string line;  
59     while(getline(fin,line)) {  
60         if(line.empty()) continue;  
61         auto parts = split(line, '|');  
62         if(parts.size() < 9) continue;  
63         Route r;  
64         r.route_id = stoi(parts[0]);  
65         r.type = parts[1];  
66         r.source = parts[2];  
67         r.destination = parts[3];  
68         r.date = parts[4];  
69         r.time = parts[5];  
70         r.fare = stod(parts[6]);  
71         r.total_seats = stoi(parts[7]);  
72         r.booked_seats = stoi(parts[8]);  
73         routes.push_back(r);  
74     }  
75     fin.close();  
76 }  
77  
78 void saveRoutes(const string &filename="routes.txt") {  
79     ofstream fout(filename);  
80     for(auto &r: routes) {  
81         fout << r.route_id << '|' << r.type << '|' << r.source << '|' << r.destination << '|'  
82         << r.date << '|' << r.time << '|' << r.fare << '|' << r.total_seats << '|' << r.  
83         booked_seats << '\n';  
84     }  
85 }
```

The status bar at the bottom indicates the current file has 218 lines and 20 columns, with 4 spaces per line, using UTF-8 encoding, and is a C++ file.

The screenshot shows the Visual Studio Code (VS Code) interface. The title bar reads "File Edit Selection View Go Run ...". The search bar contains "c++coding". The top right corner has standard window controls. On the left is the Explorer sidebar with a tree view of files and folders. The "C++ CODING" section is expanded, showing ".vscode", "project\_mini.cpp", "mini\_project.cpp", "mini\_project.exe", "mini\_project2.cpp", "settings.json", "tasks.json", and "practice.cpp". The file "inheritance.cpp" is currently selected and highlighted in yellow. The main editor area displays the "mini\_project2.cpp" file. The code is as follows:

```
79 void saveRoutes(const string &filename="routes.txt") {
80     for(auto &r: routes) {
81         fcout << r;
82     }
83     fcout.close();
84 }
85
86 }
87
88 void loadBookings(const string &filename="bookings.txt") {
89     bookings.clear();
90     ifstream fin(filename);
91     if(!fin) return;
92     string line;
93     while(getline(fin,line)) {
94         if(line.empty()) continue;
95         auto parts = split(line, '|');
96         if(parts.size() < 7) continue;
97         Booking b;
98         b.ticket_id = stoll(parts[0]);
99         b.route_id = stoi(parts[1]);
100        b.passenger_name = parts[2];
101        b.phone = parts[3];
102        b.seats_booked = stoi(parts[4]);
103        b.seat_numbers = parts[5];
104        b.total_fare = stod(parts[6]);
105        bookings.push_back(b);
106    }
107    fin.close();
108 }
109
110 void saveBookings(const string &filename="bookings.txt") {
111     ofstream fcout(filename);
112 }
```

The status bar at the bottom indicates "Ln 218, Col 20" and "Spaces: 4" and "UTF-8". It also shows icons for C++, Win32, and Prettier.

The screenshot shows the Visual Studio Code (VS Code) interface. The title bar reads "File Edit Selection View Go Run ...". The Explorer sidebar on the left lists files and folders under "C++ CODING" and ".vscode". The ".vscode" folder contains "project\_mini.cpp", "mini\_project.cpp", "mini\_project.exe", "mini\_project2.cpp", "settings.json", "tasks.json", and "practice.cpp". The "C++ CODING" section lists several files: "inheritance.cpp" (which is currently selected), "areaofcircle.cpp", "areaofcircle.exe", "class.cpp", "class.exe", "classAndStruct.CPP", "concatenatetwosring.cpp", "concatenatetwosring.exe", "conditional.cpp", "conditional.exe", "controal\_statement.cpp", "controal\_statement.exe", "experiment1.cpp", "experiment1.exe", "firstc++.cpp", and "firstc++.exe". The main editor area displays the code for "mini\_project2.cpp". The status bar at the bottom shows "Ln 218, Col 20" and other system information.

```
110 void saveBookings(const string &filename="bookings.txt") {
111     ofstream fout(filename);
112     for(auto &b: bookings) {
113         fout << b.ticket_id << '|' << b.route_id << '|' << b.passenger_name << '|' << b.phone << '|'
114         << b.seats_booked << '|' << b.seat_numbers << '|' << b.total_fare << '\n';
115     }
116     fout.close();
117 }
118 // Generate next route id
119 int nextRouteId() {
120     int mx = 1000;
121     for(auto &r: routes) if(r.route_id > mx) mx = r.route_id;
122     return mx + 1;
123 }
124
125 // Generate ticket id (timestamp + random)
126 long long generateTicketId() {
127     auto t = chrono::system_clock::now();
128     long long ts = chrono::duration_cast<chrono::seconds>(t.time_since_epoch()).count();
129     long long rnd = rand() % 1000;
130     return ts * 1000 + rnd;
131 }
132
133 // Display routes
134 void showRoutes() {
135     if(routes.empty()) {
136         cout << "No routes available. Admin se koi route add karo.\n";
137     }
138 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** The title bar displays "File Edit Selection View Go Run ... < > c++ coding".
- File Explorer (Left Side):** The "C++ CODING" folder is expanded, showing files like project\_mini.cpp, C++ mini\_project.cpp, mini\_project.exe, C++ mini\_project2.cpp, settings.json, tasks.json, practice.cpp, inheritance.cpp, areaofcircle.cpp, areaofcircle.exe, class.cpp, class.exe, classAndStruct.CPP, concatenatestwosrings.cpp, concatenatestwosrings.exe, conditional.cpp, conditional.exe, controal\_statement.cpp, controal\_statement.exe, experiment1.cpp, experiment1.exe, firstc++.cpp, firstc++.exe, OUTLINE, and TIMELINE.
- Code Editor (Main Area):** The editor shows the content of C++ mini\_project2.cpp. The code includes functions for route management and admin functions for adding routes. The code uses standard C++ libraries and structures like vectors and streams.
- Right Panel:** A sidebar on the right displays a preview of the code or a related document.

```
135     void showRoutes() {
136         if(routes.empty()) {
137             return;
138         }
139         cout << left << setw(6) << "ID" << setw(8) << "Type" << setw(12) << "Source"
140             << setw(12) << "Dest" << setw(12) << "Date" << setw(8) << "Time"
141             << setw(8) << "Fare" << setw(10) << "SeatsAvail" << '\n';
142         cout << string(80,'-') << '\n';
143         for(auto &r: routes) {
144             cout << setw(6) << r.route_id << setw(8) << r.type << setw(12) << r.source
145                 << setw(12) << r.destination << setw(12) << r.date << setw(8) << r.time
146                 << setw(8) << r.fare << setw(10) << (r.total_seats - r.booked_seats) << '\n';
147         }
148     }
149
150 // Admin functions
151 void adminAddRoute() {
152     Route r;
153     r.route_id = nextRouteId();
154     cout << "Type (Bus/Train): "; cin >> ws; getline(cin, r.type);
155     cout << "Source: "; getline(cin, r.source);
156     cout << "Destination: "; getline(cin, r.destination);
157     cout << "Date (dd-mm-yyyy): "; getline(cin, r.date);
158     cout << "Time (hh:mm): "; getline(cin, r.time);
159     cout << "Fare (per seat): "; cin >> r.fare;
160     cout << "Total seats: "; cin >> r.total_seats;
161     r.booked_seats = 0;
162     routes.push_back(r);
163     saveRoutes();
164     cout << "Route added with ID: " << r.route_id << '\n';
165 }
```

The screenshot shows the Visual Studio Code (VS Code) interface. The title bar reads "File Edit Selection View Go Run ...". The left sidebar is the "EXPLORER" view, which lists files and folders under "C++ CODING". A file named "inheritance.cpp" is currently selected. The main editor area displays the code for "mini\_project2.cpp". The code is a C++ program with several functions: adminAddRoute(), adminDeleteRoute(), findRoute(int id), and bookTicket(). The code uses standard C++ features like vectors, iterators, and conditional statements. The status bar at the bottom shows "Ln 218, Col 20" and other settings.

```
152 void adminAddRoute() {
153     cout << "Route added with ID: " << r.route_id << "\n";
154 }
155
156 void adminDeleteRoute() {
157     int id; cout << "Enter route ID to delete: "; cin >> id;
158     auto it = remove_if(routes.begin(), routes.end(), [&](const Route &r){ return r.route_id == id; });
159     if(it != routes.end()) {
160         routes.erase(it, routes.end());
161         saveRoutes();
162         cout << "Route deleted (if existed).\n";
163     } else {
164         cout << "Route not found.\n";
165     }
166 }
167
168 // Find route by id
169 Route* findRoute(int id) {
170     for(auto &r: routes) if(r.route_id == id) return &r;
171     return nullptr;
172 }
173
174 // Booking flow
175 void bookTicket() {
176     showRoutes();
177     if(routes.empty()) return;
178     int id; cout << "Enter route ID to book: "; cin >> id;
179     Route* r = findRoute(id);
180     if(!r) { cout << "Invalid route ID.\n"; return; }
```

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The title bar includes the VS Code logo, file names (.vscode, project\_mini.cpp, mini\_project2.cpp), and a search bar with the text "c++coding". The menu bar has options like File, Edit, Selection, View, Go, Run, etc. The left sidebar contains icons for Explorer, Search, Open, Task, and Settings. The Explorer view shows a tree structure for a C++ project named "C++CODING" under ".vscode". The files listed include project\_mini.cpp, mini\_project.cpp, mini\_project.exe, mini\_project2.cpp, settings.json, tasks.json, practice.cpp, inheritance.cpp, areaofcircle.cpp, areaofcircle.exe, class.cpp, class.exe, classAndStruct.CPP, concatenateTwosring.cpp, concatenateTwosring.exe, conditional.cpp, conditional.exe, controal\_statement.cpp, controal\_statement.exe, experiment1.cpp, experiment1.exe, firstc++.cpp, firstc++.exe, OUTLINE, and TIMELINE. The main editor area displays the code for "mini\_project2.cpp". The code is a function named bookTicket() that prompts the user for a route ID, checks availability, and books seats. It uses std::vector for seat numbers and std::string for seat numbers. The status bar at the bottom shows "Ln 218, Col 20" and other system information.

```
File Edit Selection View Go Run ... ← → 🔍 c++coding
EXPLORER
C++CODING
.vscode > project_mini.cpp > C++ mini_project2.cpp
187 void bookTicket() {
190     int id; cout << "Enter route ID to book: "; cin >> id;
191     Route* r = findRoute(id);
192     if(!r) { cout << "Invalid route ID.\n"; return; }
193     int avail = r->total_seats - r->booked_seats;
194     cout << "Available seats: " << avail << "\n";
195     if(avail <= 0) { cout << "No seats left.\n"; return; }
196     int seats; cout << "How many seats to book? "; cin >> seats;
197     if(seats <=0 || seats > avail) { cout << "Invalid number of seats.\n"; return; }
198     cin.ignore(numeric_limits<streamsize>::max(), '\n');
199     Booking b;
200     b.ticket_id = generateTicketId();
201     b.route_id = r->route_id;
202     cout << "Passenger name: "; getline(cin, b.passenger_name);
203     cout << "Phone number: "; getline(cin, b.phone);
204
205     // For simplicity seat numbers = booked_seats+1 ... +seats
206     vector<int> seatnums;
207     int start = r->booked_seats + 1;
208     for(int i=0;i<seats;i++) seatnums.push_back(start + i);
209     string sn;
210     for(size_t i=0;i<seatnums.size();++i) {
211         sn += to_string(seatnums[i]);
212         if(i+1<seatnums.size()) sn += ",";
213     }
214     b.seat_numbers = sn;
215     b.seats_booked = seats;
216     b.total_fare = seats * r->fare;
217 }
```

Ln 218, Col 20 Spaces: 4 UTF-8 CRLF {} C++ Win32

A screenshot of the Visual Studio Code (VS Code) interface. The window title is "File Edit Selection View Go Run ...". The status bar at the bottom shows "Ln 218, Col 20 Spaces: 4 UTF-8 CRLF {} C++ Win32 ⚙️ Prettier".

The Explorer sidebar on the left lists several C++ files and projects:

- project\_mini.cpp
- mini\_project.cpp
- mini\_project.exe
- mini\_project2.cpp
- settings.json
- tasks.json
- practice.cpp
- inheritance.cpp (highlighted)
- areaofcircle.cpp
- areaofcircle.exe
- class.cpp
- class.exe
- classAndStruct.CPP
- concatenatestwosring.cpp
- concatenatestwosring.exe
- conditional.cpp
- conditional.exe
- controal\_statement.cpp
- controal\_statement.exe
- experiment1.cpp
- experiment1.exe
- firstc++.cpp
- firstc++.exe

The main code editor area displays the content of `mini_project2.cpp`:

```
187 void bookTicket() {
188     // Update route
189     r->booked_seats += seats;
190     bookings.push_back(b);
191     saveRoutes();
192     saveBookings();
193     cout << "Booking successful! Ticket ID: " << b.ticket_id << "\n";
194     cout << "Seats: " << b.seat_numbers << " | Total Fare: " << b.total_fare << "\n";
195 }
196
197 void cancelTicket() {
198     long long tid; cout << "Enter Ticket ID to cancel: "; cin >> tid;
199     auto it = find_if(bookings.begin(), bookings.end(), [&](const Booking &b){ return b.ticket_id == tid; });
200     if(it == bookings.end()) { cout << "Ticket not found.\n"; return; }
201     // free seats - reduce booked_seats
202     Route* r = findRoute(it->route_id);
203     if(r) {
204         r->booked_seats -= it->seats_booked;
205         if(r->booked_seats < 0) r->booked_seats = 0;
206     }
207     cout << "Booking for " << it->passenger_name << " cancelled. Seats freed: " << it->seat_numbers
208     << "\n";
209     bookings.erase(it);
210     saveRoutes();
211     saveBookings();
212 }
213
214 void viewMyBookings() {
```

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Q c++coding
- Code Editor:** The active file is `mini_project2.cpp`. The code implements a function `viewMyBookings()` that searches for bookings based on either a phone number or a ticket ID.
- Explorer:** The sidebar shows the project structure under `C++ CODING`, including files like `project_mini.cpp`, `mini_project.cpp`, `mini_project.exe`, `mini_project2.cpp`, `settings.json`, `tasks.json`, and `practice.cpp`. The file `inheritance.cpp` is currently selected.
- Bottom Status Bar:** Lines 218, Column 20, Spaces: 4, UTF-8, CRLF, C++, Win32, Prettier

```
243     void viewMyBookings() {  
244         cout << "Search by (1) Phone (2) Ticket ID : ";  
245         int ch; cin >> ch;  
246         if(ch==1) {  
247             string ph; cout << "Enter phone: "; cin >> ph;  
248             bool found = false;  
249             for(auto &b: bookings) {  
250                 if(b.phone == ph) {  
251                     cout << "TicketID: " << b.ticket_id << " | RouteID: " << b.route_id << " | Name: "  
252                     << b.passenger_name  
253                     << " | Seats: " << b.seat_numbers << " | Fare: " << b.total_fare << '\n';  
254                     found = true;  
255                 }  
256                 if(!found) cout << "No bookings found for this phone.\n";  
257             } else if(ch==2) {  
258                 long long tid; cout << "Enter ticket id: "; cin >> tid;  
259                 bool found=false;  
260                 for(auto &b: bookings) {  
261                     if(b.ticket_id == tid) {  
262                         cout << "TicketID: " << b.ticket_id << " | RouteID: " << b.route_id << " | Name: "  
263                         << b.passenger_name  
264                         << " | Phone: " << b.phone << " | Seats: " << b.seat_numbers << " | Fare: " <<  
265                         b.total_fare << '\n';  
266                         found=true;  
267                     }  
268                     if(!found) cout << "Ticket not found.\n";  
269                 } else {  
270                     cout << "Invalid choice.\n";  
271                 }  
272             }  
273         }  
274     }
```

The screenshot shows the Visual Studio Code (VS Code) interface. The left sidebar has icons for Explorer, Search, Problems, and others. The Explorer view shows a C++ Coding folder containing .vscode, project\_mini.cpp, mini\_project.cpp, mini\_project.exe, mini\_project2.cpp, settings.json, tasks.json, practice.cpp, and inheritance.cpp. inheritance.cpp is currently selected. The main editor area displays the code for mini\_project2.cpp, which includes functions for viewing bookings, managing routes, and the main menu. The status bar at the bottom shows file statistics like 'Ln 218, Col 20' and encoding information like 'UTF-8'.

```
File Edit Selection View Go Run ... ← → Q c++coding
EXPLORER
C++ CODING
.vscode
project_mini.cpp
mini_project.cpp
mini_project.exe
mini_project2.cpp
{} settings.json
{} tasks.json
practice.cpp
inheritance.cpp
areaofcircle.cpp
areaofcircle.exe
class.cpp
class.exe
classAndStruct.CPP
concatenatetwosring.cpp
concatenatetwosring.exe
conditional.cpp
conditional.exe
controal_statement.cpp
contral_statement.exe
experiment1.cpp
experiment1.exe
firstc++.cpp
firstc++.exe
OUTLINE
TIMELINE
0 △ 0
```

```
mini_project2.cpp X
243 void viewMyBookings() {
268 } else {
270 }
271 }

void adminMenu() {
274 while(true) {
275     cout << "\n--- Admin Menu ---\n";
276     cout << "1. Add Route\n2. Delete Route\n3. View Routes\n4. Back to Main\nChoose: ";
277     int c; cin >> c;
278     cin.ignore(numeric_limits<streamsize>::max(), '\n');
279     if(c==1) adminAddRoute();
280     else if(c==2) adminDeleteRoute();
281     else if(c==3) showRoutes();
282     else break;
283 }
284 }

int main() {
287     srand(time(nullptr));
288     loadRoutes();
289     loadBookings();
290     cout << "===== Bus/Train Booking System =====\n";
291     while(true) {
292         cout << "\nMain Menu:\n1. View Routes\n2. Book Ticket\n3. Cancel Ticket\n4. View My
293 Bookings\n5. Admin Login\n6. Exit\nChoose option: ";
294         int opt; cin >> opt;
295         cin.ignore(numeric_limits<streamsize>::max(), '\n');
296         if(opt==1) {
            showRoutes();
```

Ln 218, Col 20 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** mini\_project2.cpp
- Code Editor:** The main area displays the content of `mini_project2.cpp`. The code handles user input for route operations and admin password verification.

```
int opt; cin >> opt;
if(opt==1) {
    showRoutes();
} else if(opt==2) {
    bookTicket();
} else if(opt==3) {
    cancelTicket();
} else if(opt==4) {
    viewMyBookings();
} else if(opt==5) {
    // simple admin password (for demo)
    string pass;
    cout << "Enter admin password: ";
    getline(cin, pass);
    // default password = admin123 (you can change)
    if(pass == "admin123") {
        adminMenu();
    } else {
        cout << "Incorrect password.\n";
    }
} else if(opt==6) {
    cout << "Exiting. Bye!\n";
    break;
} else {
    cout << "Invalid option.\n";
}
return 0;
```

- Explorer Bar:** Shows a list of C++ files in the project, including `project_mini.cpp`, `mini_project.cpp`, `mini_project2.cpp`, `settings.json`, `tasks.json`, and several practice files like `inheritance.cpp`, `areafcircle.cpp`, etc.
- Status Bar:** Ls 218, Col 20, Spaces: 4, UTF-8, CRLF, {}, C++, Win32, Prettier