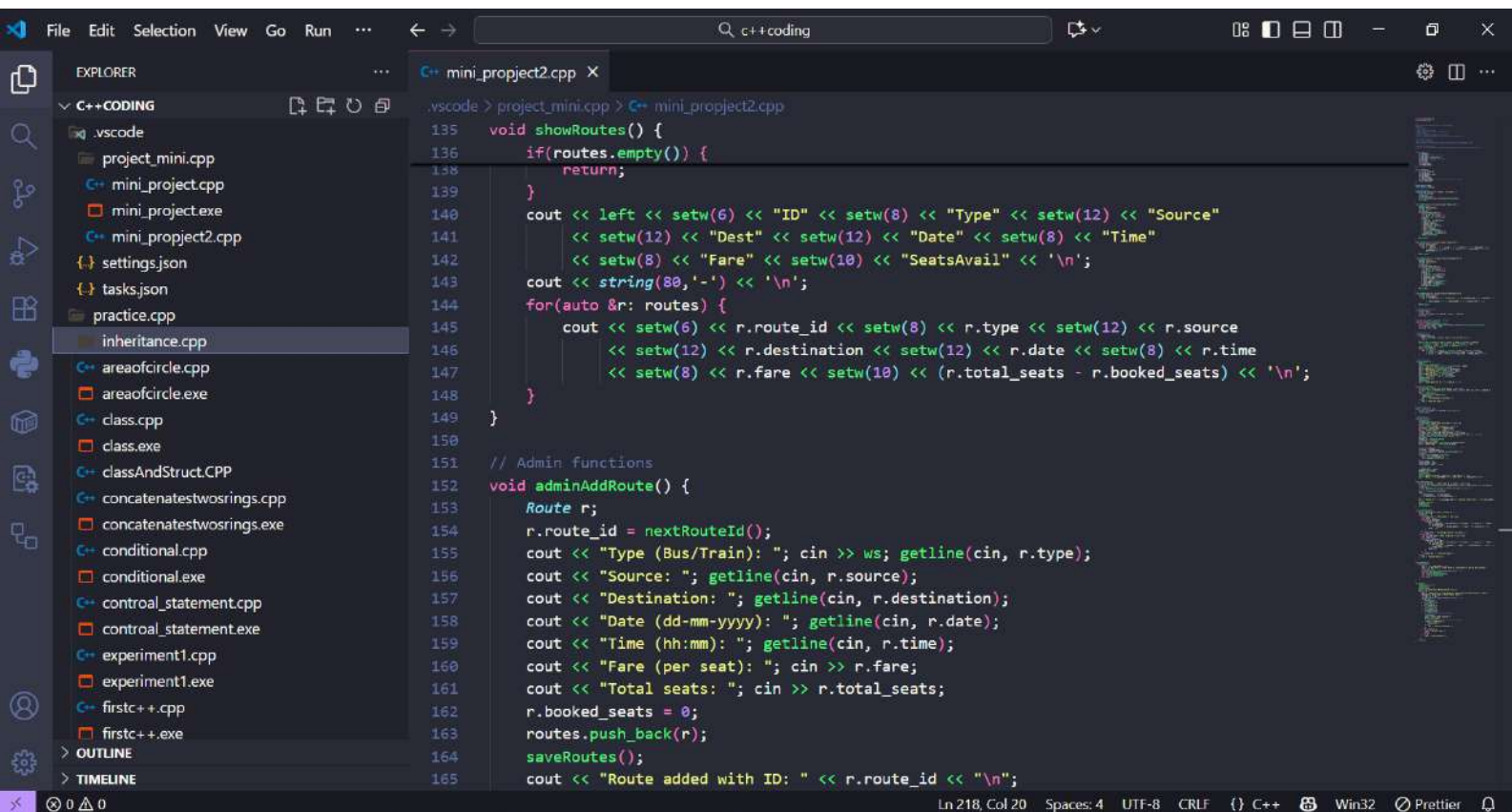


The image shows a screenshot of the Visual Studio Code editor interface. The Explorer panel on the left displays a file tree for a C++ project. The main editor window shows the code for `mini_project2.cpp`. The code includes functions for saving bookings, generating the next route ID, generating a ticket ID, and displaying routes. The status bar at the bottom indicates the current line and column (Ln 218, Col 20), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the language (C++), the platform (Win32), and the formatter (Prettier).

```
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
119 // Generate next route id
120 int nextRouteId() {
121     int mx = 1000;
122     for(auto &r: routes) if(r.route_id > mx) mx = r.route_id;
123     return mx + 1;
124 }
125
126 // Generate ticket id (timestamp + random)
127 long long generateTicketId() {
128     auto t = chrono::system_clock::now();
129     long long ts = chrono::duration_cast<chrono::seconds>(t.time_since_epoch()).count();
130     long long rnd = rand() % 1000;
131     return ts * 1000 + rnd;
132 }
133
134 // Display routes
135 void showRoutes() {
136     if(routes.empty()) {
137         cout << "No routes available. Admin se koi route add karo.\n";
138     }
139 }
```

```
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 image shows a Visual Studio Code editor window with a C++ project. The Explorer sidebar on the left lists files in a folder named 'C++CODING'. The main editor displays the file 'mini_project2.cpp' with the following code:

```
.vscode > project_mini.cpp > C++ mini_project2.cpp
152 void adminAddRoute() {
165     cout << "Route added with ID: " << r.route_id << "\n";
166 }
167
168 void adminDeleteRoute() {
169     int id; cout << "Enter route ID to delete: "; cin >> id;
170     auto it = remove_if(routes.begin(), routes.end(), [&](const Route &r){ return r.route_id ==
171     id; });
172     if(it != routes.end()) {
173         routes.erase(it, routes.end());
174         saveRoutes();
175         cout << "Route deleted (if existed).\n";
176     } else {
177         cout << "Route not found.\n";
178     }
179 }
180 // Find route by id
181 Route* findRoute(int id) {
182     for(auto &r: routes) if(r.route_id == id) return &r;
183     return nullptr;
184 }
185
186 // Booking flow
187 void bookTicket() {
188     showRoutes();
189     if(routes.empty()) return;
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; }
```

The status bar at the bottom indicates the current position is Line 218, Column 20, with 4 spaces, UTF-8 encoding, CRLF line endings, C++ language, Win32 platform, and Prettier formatting.

```
.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
218     // Update route
```

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

File Edit Selection View Go Run ...

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
 - concatenatestwsrings.cpp
 - concatenatestwsrings.exe
 - conditional.cpp
 - conditional.exe
 - control_statement.cpp
 - control_statement.exe
 - experiment1.cpp
 - experiment1.exe
 - firstc++.cpp
 - firstc++.exe
 - OUTLINE
 - TIMELINE

mini_project2.cpp

```
.vscode > project_mini.cpp > mini_project2.cpp
187 void bookTicket() {
188     // Update route
218     r->booked_seats += seats;
219     bookings.push_back(b);
220     saveRoutes();
221     saveBookings();
222     cout << "Booking successful! Ticket ID: " << b.ticket_id << "\n";
223     cout << "Seats: " << b.seat_numbers << " | Total Fare: " << b.total_fare << "\n";
224 }
225
226 void cancelTicket() {
227     long long tid; cout << "Enter Ticket ID to cancel: "; cin >> tid;
228     auto it = find_if(bookings.begin(), bookings.end(), [&](const Booking &b){ return b.ticket_id
229 == tid; });
230 if(it == bookings.end()) { cout << "Ticket not found.\n"; return; }
231 // free seats - reduce booked_seats
232 Route* r = findRoute(it->route_id);
233 if(r) {
234     r->booked_seats -= it->seats_booked;
235     if(r->booked_seats < 0) r->booked_seats = 0;
236 }
237 cout << "Booking for " << it->passenger_name << " cancelled. Seats freed: " << it->seat_numbers
238 << "\n";
239 bookings.erase(it);
240 saveRoutes();
241 saveBookings();
242 }
243 void viewMyBookings() {
```

Ln 218, Col 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         }
257         if(!found) cout << "No bookings found for this phone.\n";
258     } else if(ch==2) {
259         long long tid; cout << "Enter ticket id: "; cin >> tid;
260         bool found=false;
261         for(auto &b: bookings) {
262             if(b.ticket_id == tid) {
263                 cout << "TicketID: " << b.ticket_id << " | RouteID: " << b.route_id << " | Name: "
264                     << " | Phone: " << b.phone << " | Seats: " << b.seat_numbers << " | Fare: " <<
265                     b.total_fare << '\n';
266                 found=true;
267             }
268         }
269         if(!found) cout << "Ticket not found.\n";
270     } else {
271         cout << "Invalid choice.\n";
272     }
273 }
```

```
243 void viewMyBookings() {
268 } else {
270 }
271 }
272
273 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 }
285
286 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
Bookings\n5. Admin Login\n6. Exit\nChoose option: ";
293         int opt; cin >> opt;
294         cin.ignore(numeric_limits<streamsize>::max(), '\n');
295         if(opt==1) {
296             showRoutes();
```

```
294 int opt; cin >> opt;
295 cin.ignore(numeric_limits<streamsize>::max(), '\n');
296 if(opt==1) {
297     showRoutes();
298 } else if(opt==2) {
299     bookTicket();
300 } else if(opt==3) {
301     cancelTicket();
302 } else if(opt==4) {
303     viewMyBookings();
304 } else if(opt==5) {
305     // simple admin password (for demo)
306     string pass;
307     cout << "Enter admin password: ";
308     getline(cin, pass);
309     // default password = admin123 (you can change)
310     if(pass == "admin123") {
311         adminMenu();
312     } else {
313         cout << "Incorrect password.\n";
314     }
315 } else if(opt==6) {
316     cout << "Exiting. Bye!\n";
317     break;
318 } else {
319     cout << "Invalid option.\n";
320 }
321 return 0;
322 }
```