Code 2: TicketBookingSystem

```
#include <iostream>
#include <vector>
#include <string>
#include <exception>

using namespace std;

// Base Exception
class RailwayException : public exception {
protected:
   string message;

public:
   RailwayException(string msg) : message(msg) {}

   const char* what() const noexcept override {
       return message.c_str();
   }
};

// Derived Exceptions
class TrainFullException : public RailwayException {
public:
   TrainFullException() : RailwayException("Train is full. No seats
available.") {}
};

class InvalidSeatNumberException : public RailwayException {
public:
   InvalidSeatNumberException() : RailwayException("Invalid seat number.")
{}
};
```

```
class BookingFailedException : public RailwayException {
public:
   BookingFailedException() : RailwayException("Ticket booking failed.")
{}
};

class CancellationFailedException : public RailwayException {
public:
   CancellationFailedException() : RailwayException("Ticket cancellation
failed.") {}
};

// Train Class
class Train {
   vector<string> seats;
   int capacity;

public:
   Train(int cap) : capacity(cap) {
       seats.resize(capacity, "EMPTY");   // Resource allocation (RAII)
   }

   void bookTicket(string passengerName, int seatNumber) {
       if (seatNumber < 1 || seatNumber > capacity) {
           throw InvalidSeatNumberException();
       }

       if (seats[seatNumber - 1] != "EMPTY") {
           throw BookingFailedException();
       }

       bool available = false;
       for (string s : seats) {
           if (s == "EMPTY") {
               available = true;
               break;
```

```
            }
        }

        if (!available) {
            throw TrainFullException();
        }

        seats[seatNumber - 1] = passengerName;
        cout << "Ticket booked successfully for " << passengerName << endl;
    }

    void cancelTicket(string passengerName) {
        for (int i = 0; i < capacity; i++) {
            if (seats[i] == passengerName) {
                seats[i] = "EMPTY";
                cout << "Ticket cancelled successfully for " <<
passengerName << endl;
                return;
            }
        }
        throw CancellationFailedException();
    }

    void displaySeats() {
        cout << "\nSeat Status:\n";
        for (int i = 0; i < capacity; i++) {
            cout << "Seat " << i + 1 << " : " << seats[i] << endl;
        }
    }
};

// Main Function
int main() {
    Train train(10);

    try {
```

```
        try {
            train.bookTicket("Anuj", 1);
            train.bookTicket("Rahul", 2);
            train.bookTicket("Amit", 3);
            train.bookTicket("Neha", 4);
            train.bookTicket("Baba", 5);
            train.bookTicket("Tiwari", 6);
        }catch (InvalidSeatNumberException& e) {
            cout << "Error InvalidSeatNumber: " << e.what() << endl;
        }

        try {
            train.bookTicket("Ravi", 1);   // Seat already booked
        }catch (BookingFailedException& e) {
            cout << "Error Booking Failed: " << e.what() << endl;
        }

        try {
            train.cancelTicket("Ramu");  // Not booked
        }catch (CancellationFailedException& e) {
            cout << "Error: " << e.what() << endl;
        }

        try {
            train.bookTicket("Hari", 7);  // Not booked
        }catch (TrainFullException& e) {
            cout << "Error: " << e.what() << endl;
        }
    }
    catch (RailwayException& e) {
        cout << "Railway Exception: " << e.what() << endl;
    }

    train.displaySeats();

    return 0;
```

```
}
```

Output:

```
ha'kuna ma'tata: cd "/Users/anujgahra/Developer/PG-DAC/C++/ModuleE
Developer/PG-DAC/C++/ModuleExamC++/"TrainBookingSystem
Ticket booked successfully for Rahul
Ticket booked successfully for Amit
Ticket booked successfully for Neha
Ticket booked successfully for Baba
Ticket booked successfully for Tiwari
Error Booking Failed: Ticket booking failed.
Error: Ticket cancellation failed.
Ticket booked successfully for Hari

Seat Status:
Seat 1 : Anuj
Seat 2 : Rahul
Seat 3 : Amit
Seat 4 : Neha
Seat 5 : Baba
Seat 6 : Tiwari
Seat 7 : Hari
Seat 8 : EMPTY
Seat 9 : EMPTY
Seat 10 : EMPTY
ha'kuna ma'tata:
```