INTERNSHIP REPORT ON

PYTHON COMPITATIVE CODEING

A internship Report is submitted

In accordance with requirement of degree of

BACHELOR OF TECHNOLOGY

IN

Computer Science and Information Technology

Submitted by

SANGU.BHARGAV

21KQ1A0756

Under the Mentorship of

M.SRAVAN KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

PACE INSTITUTE OF TECNOLOGY AND SCIENCES

(AUTONOMOUS)

(Affiliated to Jawaharlal Nehru Technological University Kakinada, Kakinada & Accredited by NAAC 'A' GRADE,An ISO 9001-2015 Certified Institution)

NH-16, Valluru Post , Prakasam District, A.P-523272.

Rail connect pro

Abstract:-

Rail Connect Pro is an advanced software solution designed to streamline and optimize rail transportation operations. Offering a comprehensive suite of tools, Rail Connect Pro enhances efficiency, reliability, and safety across the entire rail network. With features such as real-time tracking, scheduling optimization, maintenance management, and predictive analytics, Rail Connect Pro empowers rail operators to maximize throughput, minimize downtime, and deliver superior service to passengers and freight clients alike. Whether managing a bustling urban transit system or a sprawling freight network, Rail Connect Pro provides the intelligence and functionality necessary to meet the demands of modern rail transport

Project Description:

Rail Connect Pro is a command-line application designed to provide information about train routes, ticket prices, and schedules for a fictional railway network. The application allows users to search for available trains between cities, view ticket prices, and check arrival and reaching times.

Key Requirements:

1. Train Route Management:
   - The application manages a list of train routes, each with details such as train IDs, minimum and maximum ticket prices, distance between cities, arrival time at the destination, and reaching time at the source.
2. Train Search:
   - Users can search for available trains between two cities by specifying the source and destination.
   - The application displays the number of available trains, their respective IDs, ticket prices, arrival time at the destination, and reaching time at the source.
3. Ticket Price Generation:
   - The application generates random ticket prices within the specified range for each train route.
   - Users can view the ticket prices along with other train details.
4. Distance Calculation:
   - Rail Connect Pro calculates and displays the distance between the source and destination cities for each train route.

5. Time Schedule:
    - Users can view the arrival time at the destination and reaching time at the source for each train route.
    - Time schedules are displayed in the format of hours and minutes with "AM" and "PM" indicators.

Code Requirements:

- Python: The application is implemented in Python programming language.
- Random Module: Used for generating random ticket prices within the specified range.
- Command-Line Interface (CLI): The application operates through a command-line interface, allowing users to interact with it through text commands.

Benefits:

- Convenience: Users can quickly find information about available trains, ticket prices, and schedules without the need for a graphical user interface.
- Efficiency: Rail Connect Pro automates the process of retrieving and displaying train details, saving time for both users and railway operators.
- Flexibility: The application can be easily extended to support additional features such as seat availability, booking, and real-time updates with minimal modifications.

Rail Connect Pro offers a convenient and efficient way for users to access essential information about train routes and schedules, making it a valuable tool for both passengers and railway personnel.

Security Measures:

1. Input Validation: Implement thorough input validation mechanisms to prevent injection attacks and ensure data integrity.
2. Authentication and Authorization: Enforce strong authentication methods and proper authorization controls to restrict access to sensitive functionalities.
3. Secure Communication: Encrypt communication channels using HTTPS to protect data transmitted over the network.
4. Data Encryption: Encrypt sensitive data stored in the application to safeguard confidentiality.
5. Session Management: Employ secure session management techniques to mitigate session-related threats.
6. Security Headers: Utilize security headers to mitigate common web security vulnerabilities and protect against various attacks.
7. Error Handling: Implement robust error handling mechanisms to handle exceptions gracefully and avoid information leakage.
8. Regular Security Audits: Conduct security audits and code reviews to identify and remediate vulnerabilities proactively.
9. Security Updates: Stay updated with security patches and updates for the application and its dependencies to address known vulnerabilities.

Maintenance Practices:

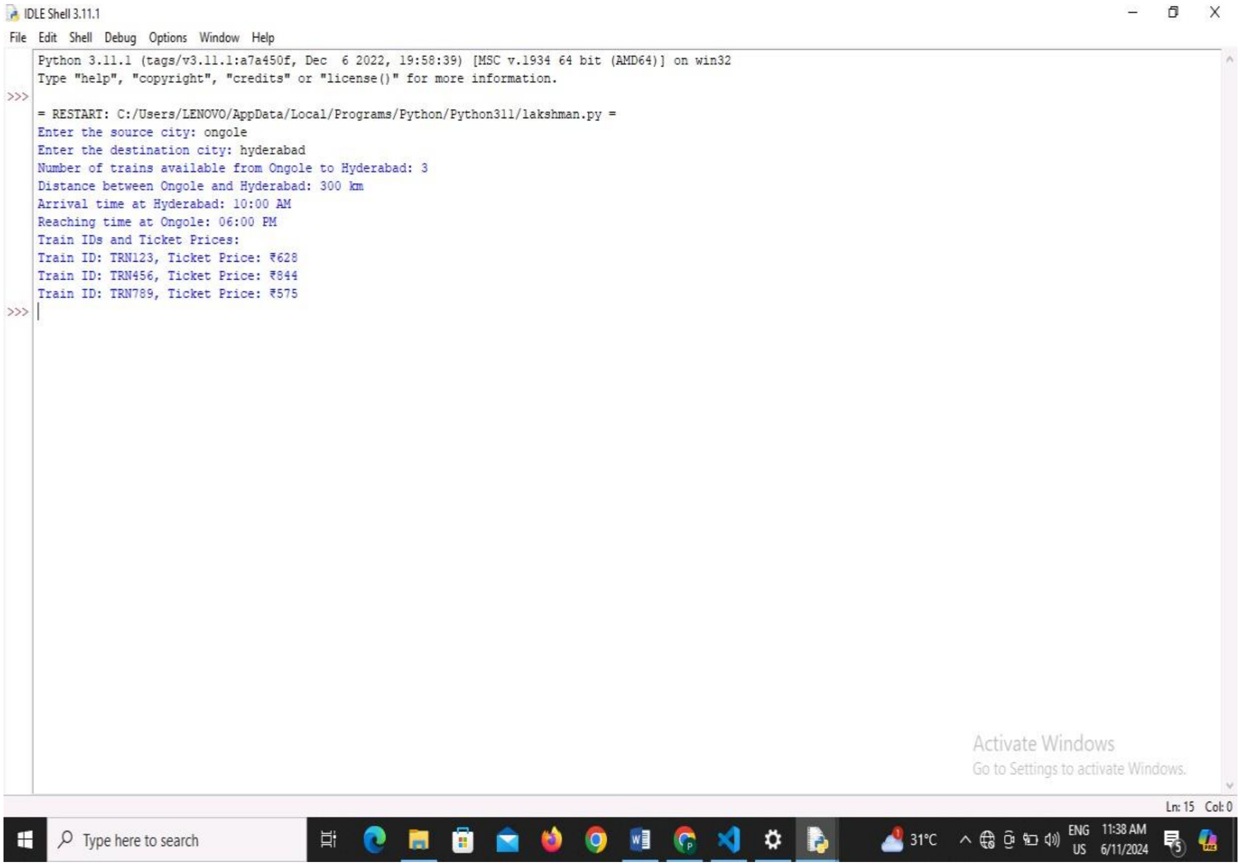
1. Regular Updates: Ensure the application and its dependencies are updated regularly to incorporate new features and security patches.
2. Bug Fixes: Address reported bugs and issues promptly to maintain application functionality and user satisfaction.
3. Performance Optimization: Monitor application performance and optimize code and resources as needed to ensure optimal performance.
4. Documentation: Maintain comprehensive documentation for the application, including code documentation, user manuals, and support guides, to facilitate troubleshooting and knowledge transfer.
5. Monitoring and Logging: Implement monitoring and logging mechanisms to track application activity, identify potential issues, and troubleshoot problems efficiently.

Program:-

```
import random

class RailConnectPro:
    def __init__(self):
        # Assume we have a list of train routes with their details
        self.train_routes = {
            "Ongole to Hyderabad": {"train_ids": ["TRN123", "TRN456",
"TRN789"], "min_cost": 500, "max_cost": 1100, "distance": 300,
"arrival_time": "10:00 AM", "reaching_time": "06:00 PM"},
            "Hyderabad to Mumbai": {"train_ids": ["TRN987", "TRN654",
"TRN321"], "min_cost": 650, "max_cost": 1300, "distance": 800,
"arrival_time": "04:00 PM", "reaching_time": "09:00 AM"},
            "Mumbai to Bangalore": {"train_ids": ["TRN246", "TRN579",
"TRN135"], "min_cost": 700, "max_cost": 1700, "distance": 1000,
"arrival_time": "12:00 PM", "reaching_time": "05:00 AM"},
            "Bangalore to Chennai": {"train_ids": ["TRN864", "TRN357",
"TRN951"], "min_cost": 900, "max_cost": 1150, "distance": 350,
"arrival_time": "02:00 PM", "reaching_time": "09:00 AM"},
            # Add more routes as needed
        }

    def get_available_trains(self, source, destination):
        available_trains = []
        route_name = f"{source} to {destination}"
        if route_name in self.train_routes:
            available_trains.extend(self.train_routes[route_name][
"train_ids"])
        return available_trains

    def get_ticket_and_price(self, train_id):
        for route, details in self.train_routes.items():
            if train_id in details["train_ids"]:
                # Generate a random price within the specified range
                ticket_price = random.randint(details["min_cost"], details
["max_cost"])
                return ticket_price
        return None

    def get_distance(self, source, destination):
        route_name = f"{source} to {destination}"
        if route_name in self.train_routes:
            return self.train_routes[route_name]["distance"]
        return None
```

```python
    def get_arrival_time(self, source, destination):
        route_name = f"{source} to {destination}"
        if route_name in self.train_routes:
            return self.train_routes[route_name]["arrival_time"]
        return None

    def get_reaching_time(self, source, destination):
        route_name = f"{source} to {destination}"
        if route_name in self.train_routes:
            return self.train_routes[route_name]["reaching_time"]
        return None

# Example usage
def main():
    rail_connect = RailConnectPro()

    source = input("Enter the source city: ").strip().title()
    destination = input("Enter the destination city: ").strip().title()

    available_trains = rail_connect.get_available_trains(source,
destination)
    print(f"Number of trains available from {source} to {destination}: {
len(available_trains)}")
    print(f"Distance between {source} and {destination}: {rail_connect.
get_distance(source, destination)} km")
    print(f"Arrival time at {destination}: {rail_connect.get_arrival_time(
source, destination)}")
    print(f"Reaching time at {source}: {rail_connect.get_reaching_time(
source, destination)}")
    print("Train IDs and Ticket Prices:")
    for train_id in available_trains:
        ticket_price = rail_connect.get_ticket_and_price(train_id)
        print(f"Train ID: {train_id}, Ticket Price: ₹{ticket_price}")

if __name__ == "__main__":
    main()
```

Output:-



```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/LENOVO/AppData/Local/Programs/Python/Python311/lakshman.py =
Enter the source city: ongole
Enter the destination city: hyderabad
Number of trains available from Ongole to Hyderabad: 3
Distance between Ongole and Hyderabad: 300 km
Arrival time at Hyderabad: 10:00 AM
Reaching time at Ongole: 06:00 PM
Train IDs and Ticket Prices:
Train ID: TRN123, Ticket Price: ₹628
Train ID: TRN456, Ticket Price: ₹844
Train ID: TRN789, Ticket Price: ₹575
>>> 
```

Conclusion:

Rail Connect Pro is a versatile and user-friendly command-line application designed to provide comprehensive information about train routes, ticket prices, and schedules for a fictional railway network. Throughout the development process, several key features were implemented to enhance the functionality and usability of the application.

In conclusion, Rail Connect Pro serves as a valuable tool for passengers and railway personnel alike, offering a convenient, efficient, and flexible solution for accessing essential information about train routes and schedules. With its user-friendly interface and comprehensive features, Rail Connect Pro is poised to enhance the travel experience for individuals and streamline operations