Capstone Project Title: Travel booking app on aws, with frontend on EC2, User data on RDS, data analysis on Glue and Athena

Objective: A travel booking app on AWS with the described architecture involves several AWS services and data analysis on Glue and Athena:

- User Travel Pattern
- Most Searched place
- Season wise travel searches
- Travel by flight vs by road

Frontend on EC2:

- EC2 instance to host your frontend application.
- Install and configure a web server (e.g., Nginx, Apache) on the EC2 instance to serve your frontend application.
- Deploy your frontend code on the EC2 instance.

User data on RDS:

- RDS instance (Relational Database Service) to store user data. Choose a database engine that fits your requirements (e.g., MySQL, PostgreSQL, or Amazon Aurora).
- Configure security groups to control the traffic to your RDS instance.
- Connect your frontend application to the RDS database for user data storage.

Data analysis on Glue and Athena:

- AWS Glue for ETL (Extract, Transform, Load) jobs to transform and prepare data for analysis. Define crawlers to discover and catalog the data in your storage.
- Store your transformed data in an Amazon S3 bucket.
- Use Athena for querying data directly in your S3 bucket using SQL queries.

- Authentication and Authorization:

- User authentication and authorization in your frontend application. You can use Amazon Cognito for user authentication and manage user pools.
- Set up appropriate IAM roles and policies to control access to other AWS services.

Scalability:

AWS Auto Scaling to automatically adjust the number of EC2 instances based on traffic.

Choose an appropriate instance type for your EC2 instances and RDS database based on your expected workload.

Monitoring and Logging:

- CloudWatch Alarms to monitor the performance of your EC2 instances, RDS instance, and other critical metrics.
- AWS CloudTrail for auditing API calls made on your AWS account.
- Configure centralized logging using services like AWS CloudWatch Logs.