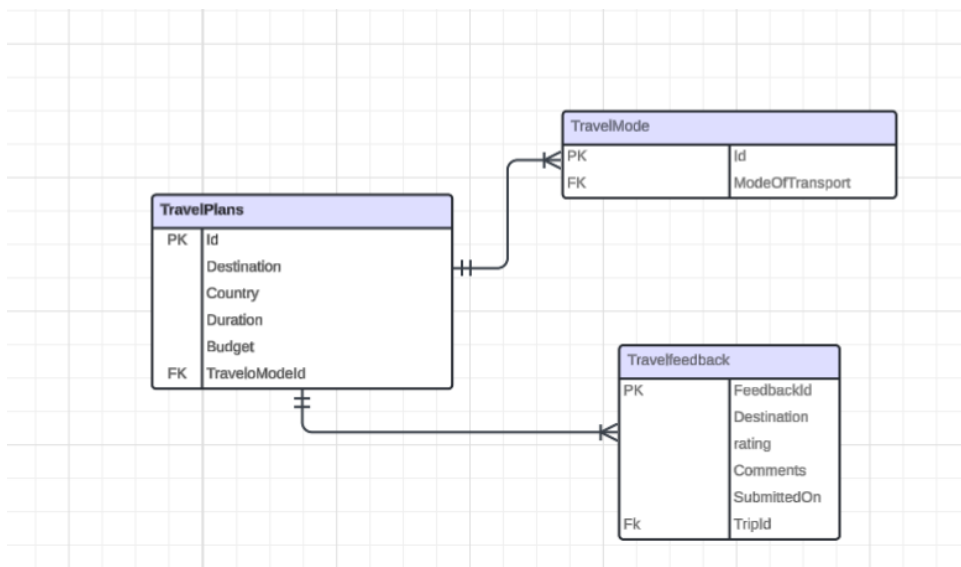


Final Project-Group 7

Deployed Application URL:

<https://tripsphereweb-e5cjgpdvajbze2e8.westeurope-01.azurewebsites.net/>

Updated ERD Diagram:



This ERD shows how data is stored and connected in a travel-related application. It has **three main tables**: TravelMode, TravelPlans, and TravelFeedback. These tables are connected using **primary keys (PK)** and **foreign keys (FK)**.

□ 1. TravelMode

- This table stores types of transport like **bus**, **train**, or **flight**.
- **Fields:**
 - Id: the unique ID of each transport (Primary Key)
 - ModeOfTransport: name of the transport (like “Flight”)
- **Relationship:**
 - One travel mode can be used in **many travel plans**.

- This is called a **one-to-many relationship**.

□ 2. TravelPlans

- This table stores information about each trip.
- **Fields:**
 - Id: the unique ID of the trip
 - Destination: place of the trip
 - Country: which country the trip is in
 - Duration: how long the trip is
 - Budget: how much the trip costs
 - TravelModeId: this links to the TravelMode table
- **Relationships:**
 - One travel mode → many travel plans
 - One travel plan → many feedback entries

□ 3. TravelFeedback

- This table stores reviews or ratings about trips.
- **Fields:**
 - FeedbackId: ID of the feedback
 - Destination: destination of the trip
 - Rating: score from 1 to 5
 - Comments: user's review
 - SubmittedOn: when the feedback was given
 - TripId: links to the trip (from TravelPlans)
- **Relationship:**
 - One trip can have **many feedback entries**.

Summary of Relationships

- One travel mode can be used in many trips.
- One trip can have many feedbacks.

Git Log Output:

https://github.com/VaishnaviChitumalla99/Tripsphere_Webapplication_Group7

Commits on May 6, 2025		
core logic	Namratharao22 authored 1 minute ago	Verified c5d8a58
adding core logic	Namratharao22 committed 2 minutes ago	148956a
adding models	skmohanty2628 authored 1 hour ago	Verified 8457d38
adding models	skmohanty2628 committed 1 hour ago	446fea2
Update README.md	skmohanty2628 committed 1 hour ago	2496be5
project setup and views addition	AnkitaTripathy35 authored 1 hour ago	Verified dad854e
project setup and views addition	AnkitaTripathy35 committed 1 hour ago	3828663
Merge pull request #3 from VaishnaviChitumalla99/feature/vaishnavi	VaishnaviChitumalla99 authored 1 hour ago	Verified a21db69
adding database	VaishnaviChitumalla99 committed 1 hour ago	69c15db
Merge pull request #2 from VaishnaviChitumalla99/feature/vaishnavi	VaishnaviChitumalla99 authored 8 hours ago	Verified 01a87d6
config file	VaishnaviChitumalla99 authored 8 hours ago	Verified ffab53a
Commits on May 5, 2025		
Update README.md	AnkitaTripathy35 authored 11 hours ago	Verified b2f1afa
Update README.md	AnkitaTripathy35 authored 11 hours ago	Verified c8bcc10
Merge pull request #1 from VaishnaviChitumalla99/Vaishnavi	VaishnaviChitumalla99 authored 11 hours ago	Verified 1ca4258
Update README.md	VaishnaviChitumalla99 authored 11 hours ago	Verified 277a75c
Update README.md	VaishnaviChitumalla99 authored 11 hours ago	Verified 38cb212
Updating README File		Verified 72a7ab9

Self-Reflection of All Team Members:

Ankita Tripathy

Time Taken on the Project:

During the project timeline, I took approximately 60 to 70 working hours in designing, developing, and perfecting the front-end of the TripSphere app. I utilized my time iteratively creating the homepage, feedback module, and analytics dashboard, making them both interactive and visually appealing.

Key Takeaways:

This project also embedded in me the MVC pattern and how to successfully link frontend pieces with backend functionality in a structured and scalable system. I learned to a large degree how to utilize CSS Grid, Flexbox, and Razor syntax. I also learned how to integrate external APIs (like the weather API), manage user interaction through form validation, and implement full CRUD operations. Importantly, I gained valuable practical experience transforming static UI layouts into fully responsive and dynamic pages.

Challenges Faced and Solutions:

Aligning the front-end presentations with backend capabilities was a top challenge, especially as new entity relationships and database fields evolved over the development timeline. To remedy this, I maintained regular contact with the backend team, updated the view models accordingly, and compartmentalized the CSS to eliminate conflicts. Enabling interactive pieces to be responsive such as embedded videos and destination galleries was also vital. I solved this using CSS media queries, flex and grid utilities for layout, and device testing.

Suggestions for Improvement:

For development in the future, using a component-based design or a UI kit (such as Bootstrap or Tailwind CSS) can quicken development and offer consistency across all pages. Incorporating user feedback early on within the UI design process would also increase usability. Last but not least, regular sprint reviews and collaborative wireframing sessions would enhance collaboration and avoid duplicate development.

Namratha Vardhini

As the Backend & Database Integration Specialist on the TripSphere project, I focused on implementing secure and efficient data flow between the application and the SQL Server database.

My responsibilities included developing the “Book Now” API to capture and store user trip details, and the “Contact Us” API to log and timestamp user messages. I worked on integrating form submissions using Entity Framework Core and ensured server-side validation to maintain data integrity.

This project helped me strengthen my understanding of the ASP.NET Core MVC architecture, especially in routing, model binding, and controller-view integration. I gained valuable experience in debugging backend issues, particularly around model mismatches and form handling. One of the challenges I faced was ensuring consistent data flow between the frontend and backend, especially when dealing with foreign key relationships like TravelMode.

If given more time, I would enhance the application by adding authentication features, expanding analytics with filter options, and implementing unit tests to improve code reliability. Overall, the project reinforced my backend development skills and deepened my understanding of building data-driven applications in a structured, real-world environment. It was a rewarding experience that allowed me to apply both technical and analytical thinking while contributing to a functional, user-focused system.

Subham Mohanty

In this project, I spent approximately 60 hours working on creating and implementing secure user interaction and authentication functionality. One of the most significant things that I learned was how to prevent common vulnerabilities such as SQL injection and cross-site scripting (XSS) attacks.

Cloud database development was another significant area of study. I worked hands-on with Azure SQL Database and used Entity Framework Core to develop and work with relational data models. On the user experience side, I created smooth authentication flows, including login, signup, and password reset, and introduced role-based access to different parts of the application. I also introduced a feature for providing feedback to make the platform interactive.

One of the largest challenges that I faced was integrating authentication in the MVC hierarchy and the API logic, which initially introduced some inconsistency. I was able to resolve this by implementing a consistent solution using JWT for API calls and cookie-based sessions for MVC pages. Another issue was securing the feedback forms. I addressed this by using proper input validation and parameterized SQL queries to prevent any malicious scripting.

In the future, the project could be improved by adding features such as OAuth 2.0 login capabilities (e.g., Google or Facebook) and enabling multi-factor authentication for high-risk actions. It would also be beneficial to include more descriptive error messages when something goes wrong, e.g., a failed login or timed-out session. Having a comprehensive developer guide would also help future teams better understand and expand upon the current system.

Overall, this project helped me to improve my skills in secure login systems and cloud data integration. It also taught me how to maintain a good balance between system security and user convenience, something that I plan to further improve in the future.

Vaishnavi Chittumalla

I am currently pursuing my Master's in Artificial Intelligence and Business Analytics at the University of South Florida. I have 2.5 years of work experience in cloud storage at Atos Global Services, and in this project, I took up the role of GitHub Manager and Documentation Lead.

I spent around 20 to 25 hours on this project. My main tasks were managing the GitHub repository, tracking version control, and writing clear documentation for the technical work we did. I also helped the team stay organized and ensured that all project files and updates were submitted properly.

Through this project, I learned how to work with ASP.NET Core MVC, how cloud deployment works using Microsoft Azure, and how to integrate public APIs into web applications. I also got more confident in working with Razor Views and Bootstrap for the front-end design.

One of the challenges we faced was handling merge conflicts when different team members worked on the same files. To solve this, I set clear GitHub rules, helped the team follow naming conventions, and made sure we all synced our changes regularly.

In future projects, I would suggest using GitHub Actions or Azure DevOps for smoother deployment and adding user login features to make the application more secure and personalized.

Presentation Slide Deck:

