

Technical Support Ticket - Object-Oriented Design

GitHub Submission Summary

GitHub Name: YourGitHubUsername

Repository Name: support-ticket-oop

Class Design Summary

User - Represents a user submitting or searching for a support ticket.

Class Variables: name (string), email (string)

Class Methods:

- constructor(name, email): Initializes a new user.
- getUserInfo(): Returns user info object.

SupportTicket - Represents a support ticket.

Class Variables: ticketId (string), user (User), issue (string), status (string), createdAt (Date)

Class Methods:

- constructor(ticketId, user, issue): Initializes the ticket.
- getSummary(): Returns a summary string.
- closeTicket(): Marks the ticket as closed.

TicketManager - Handles ticket collection and searching.

Class Variables: tickets (Array)

Class Methods:

- constructor(): Initializes an empty ticket list.
- addTicket(ticket): Adds a ticket.
- findTicketsByName(name): Finds tickets by user name.
- findTicketById(id): Finds ticket by ID.
- listOpenTickets(): Lists open tickets.

Experience Discussion

Working on this assignment helped reinforce the principles of object-oriented programming in a real-world web application. By refactoring the original functional code into class-based modules, I was able to better encapsulate data and behavior, improve code readability, and establish a clearer structure for future scalability. One of the challenges was deciding how much logic to include in the classes versus what should be handled elsewhere (like API/server interactions). I addressed this by adding comments where server calls would normally occur. This modular approach will allow seamless integration with a back-end system in the future.