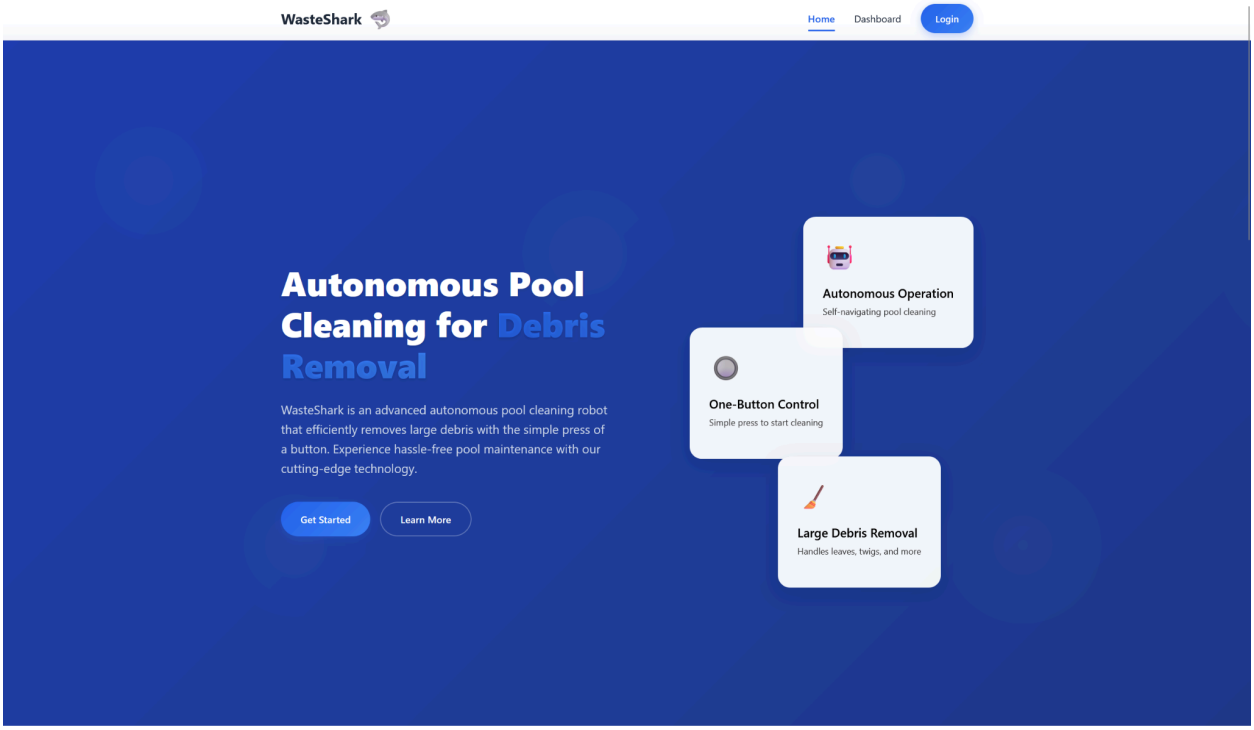
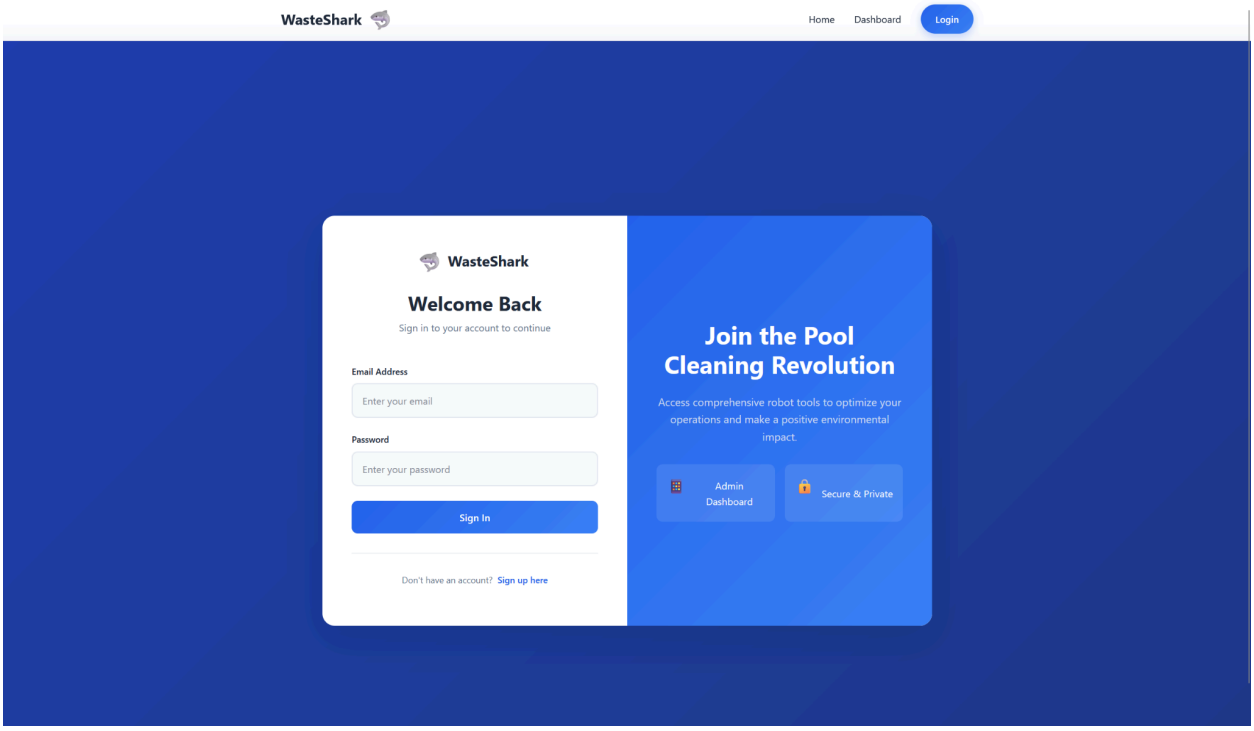
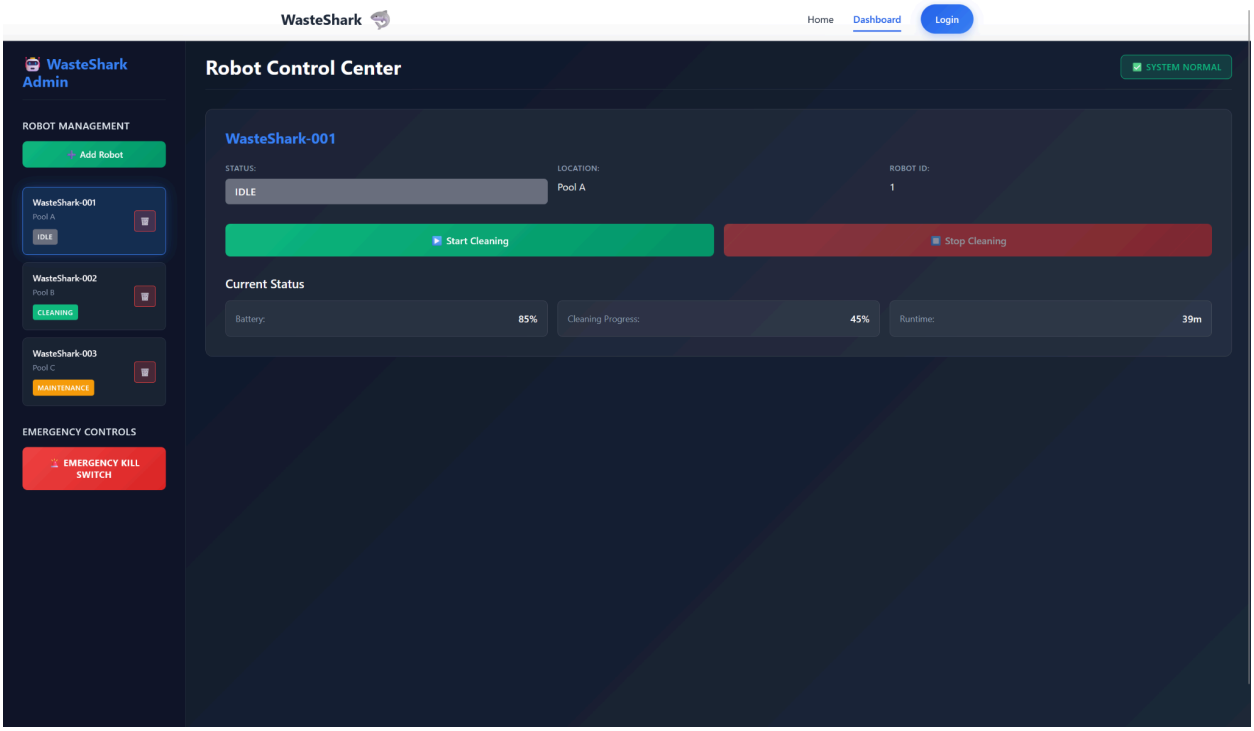
- Using **React.js + Vite**
- **Tailwind CSS for styling**

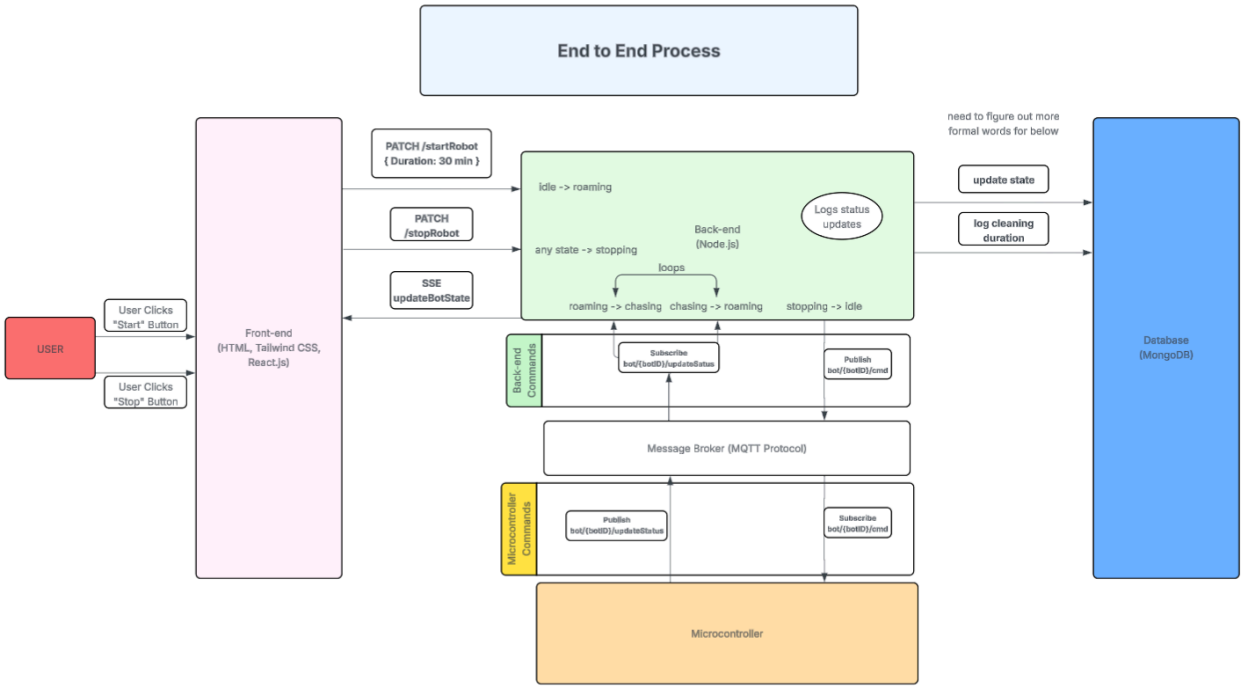


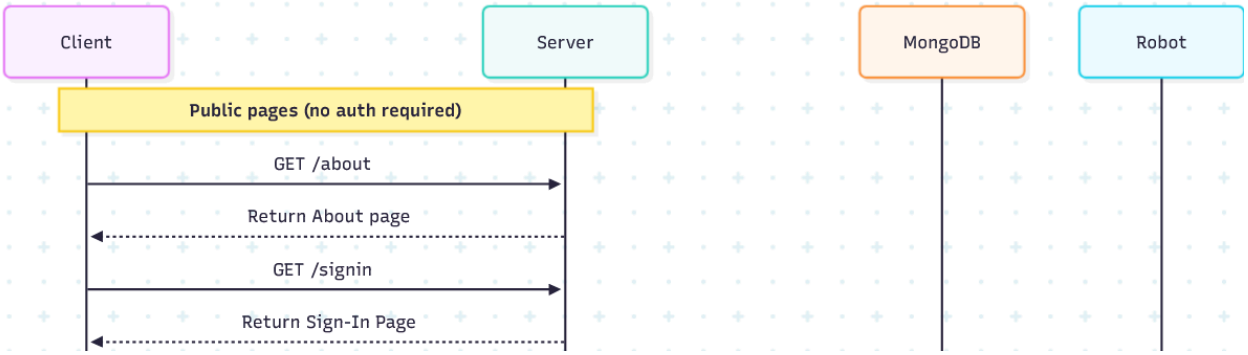


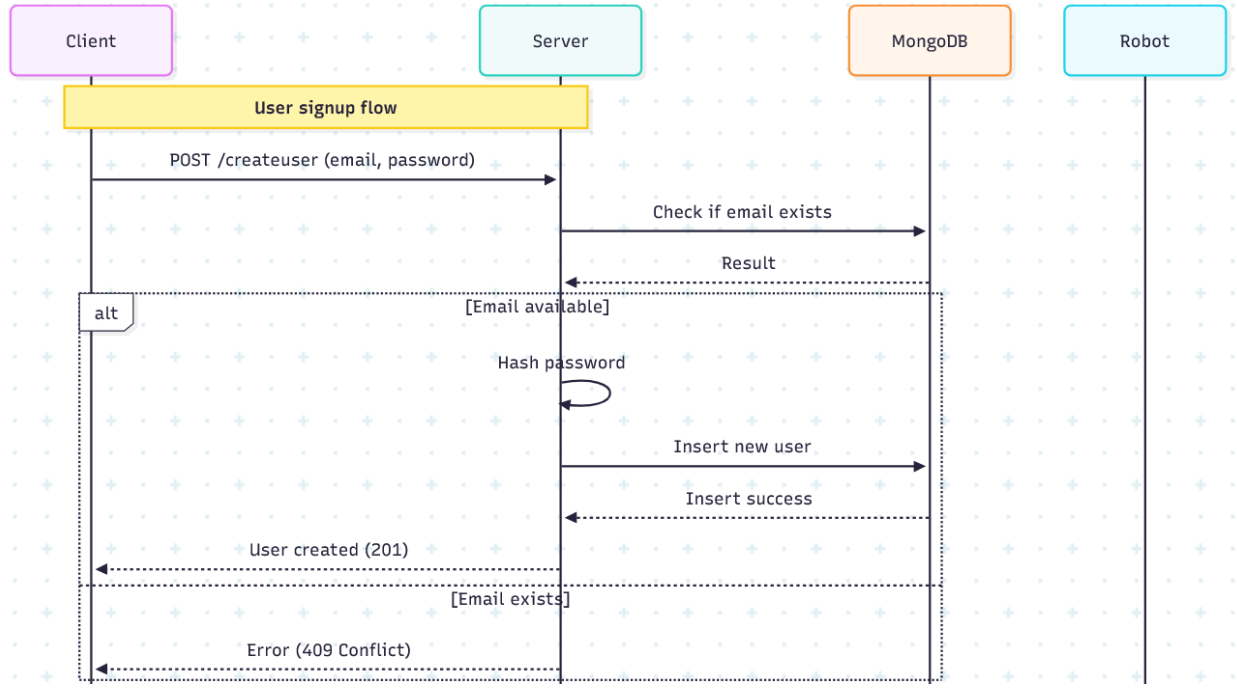
JWT for authorization and security

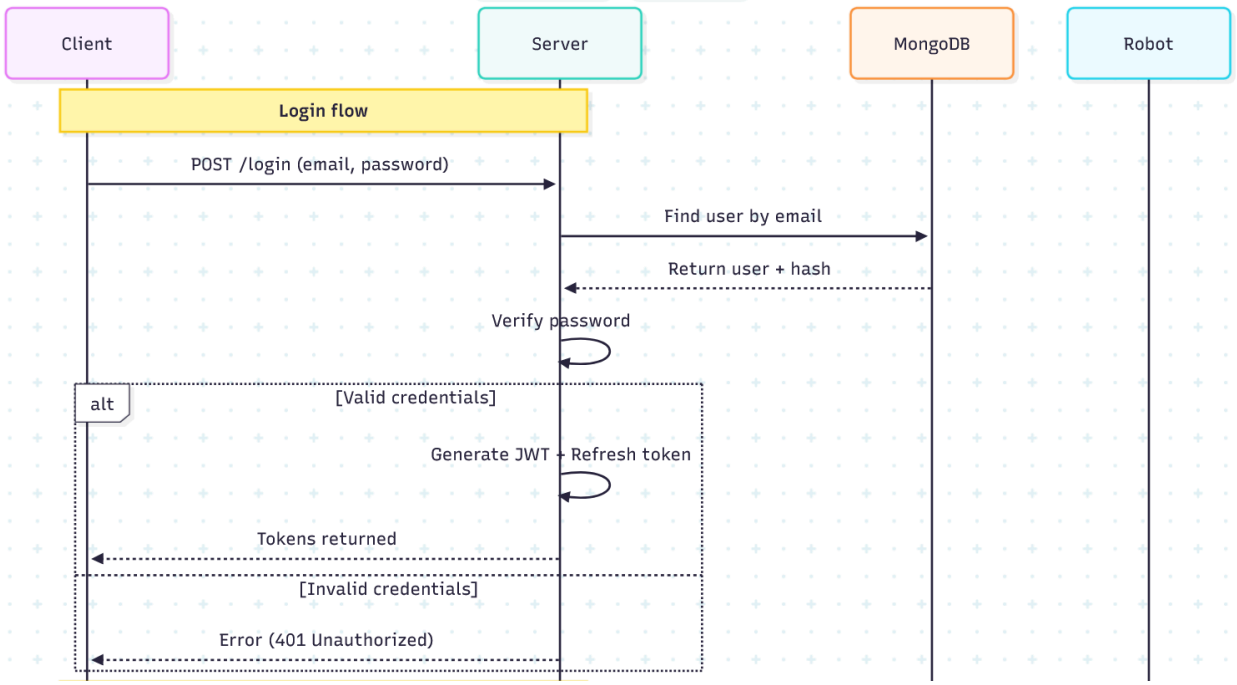


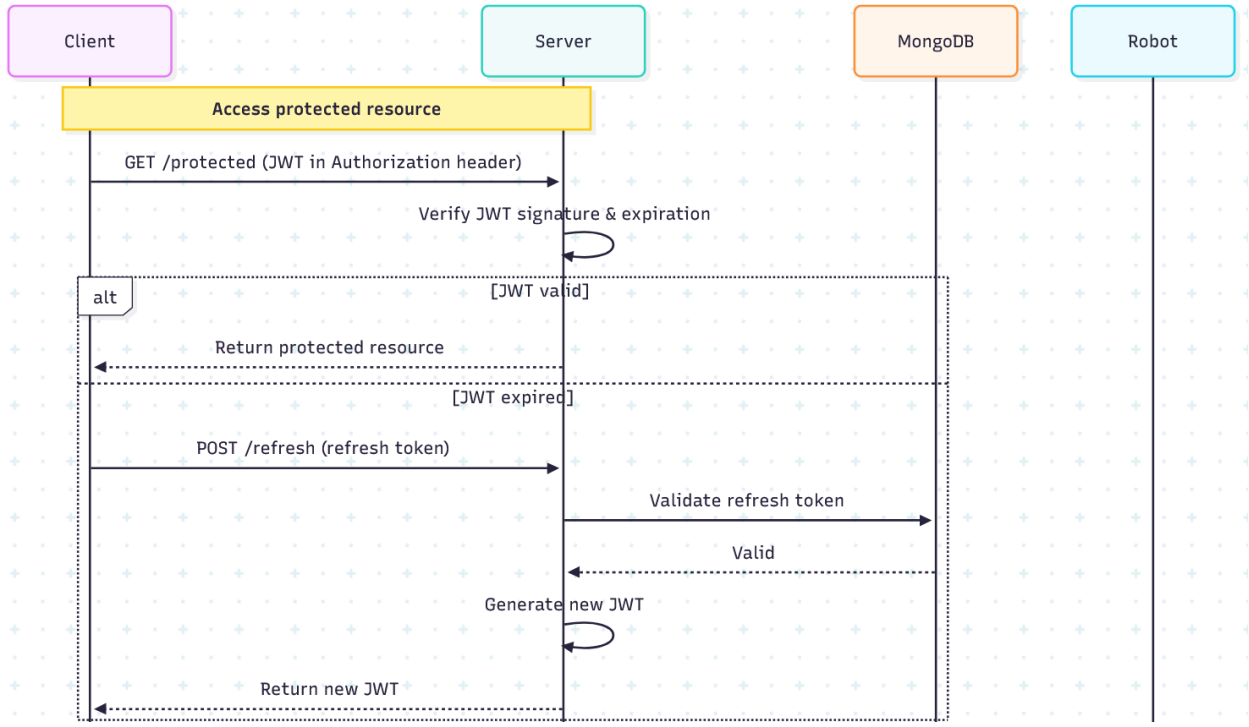
LOGO:

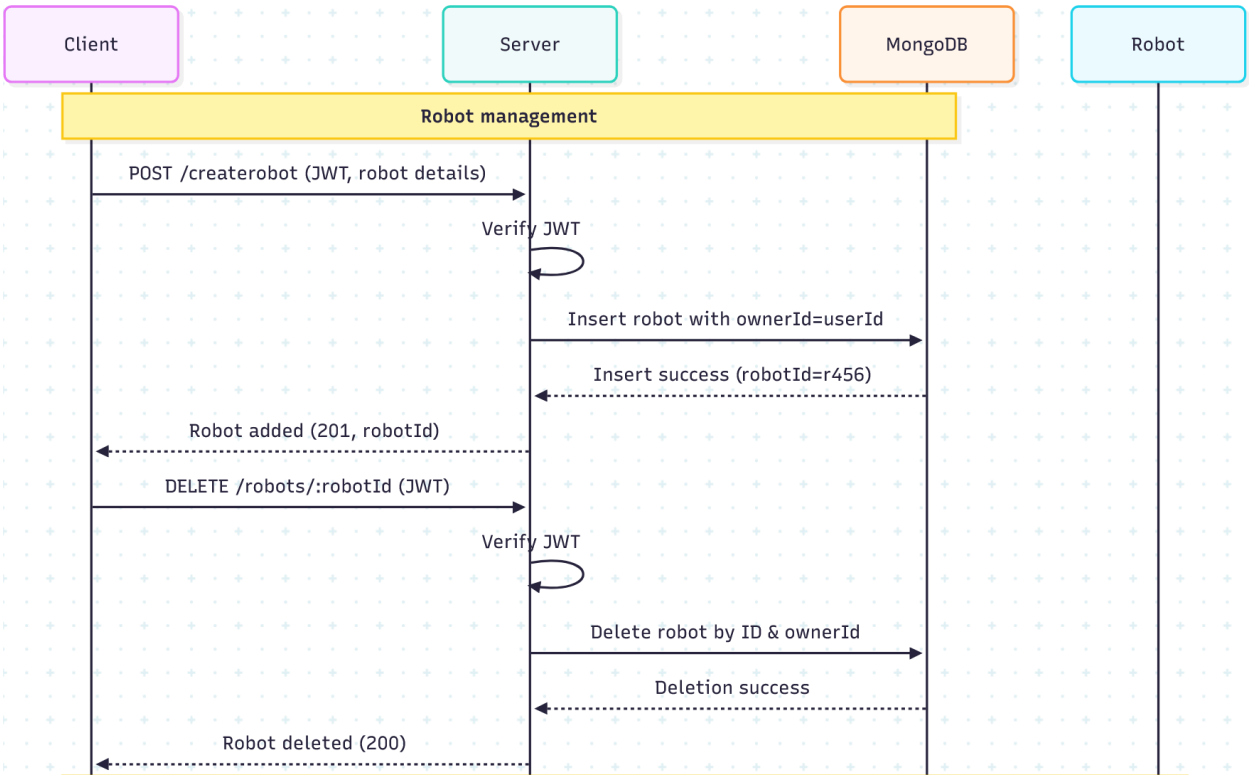

End to End Process
USER
User Clicks "Start" Button
User Clicks "Stop" Button
Front-end (HTML, Tailwind CSS, React.js)
PATCH /startRobot { Duration: 30 min }
PATCH /stopRobot
SSE updateBotState
idle -> roaming
any state -> stopping
Back-end (Node.js)
Logs status updates
loops
roaming -> chasing
chasing -> roaming
stopping -> idle
need to figure out more formal words for below
update state
log cleaning duration
Database (MongoDB)
Back-end Commands
Subscribe bot/{botID}/updateStatus
Publish bot/{botID}/cmd
Message Broker (MQTT Protocol)
Microcontroller Commands
Publish bot/{botID}/updateStatus
Subscribe bot/{botID}/cmd
Microcontroller

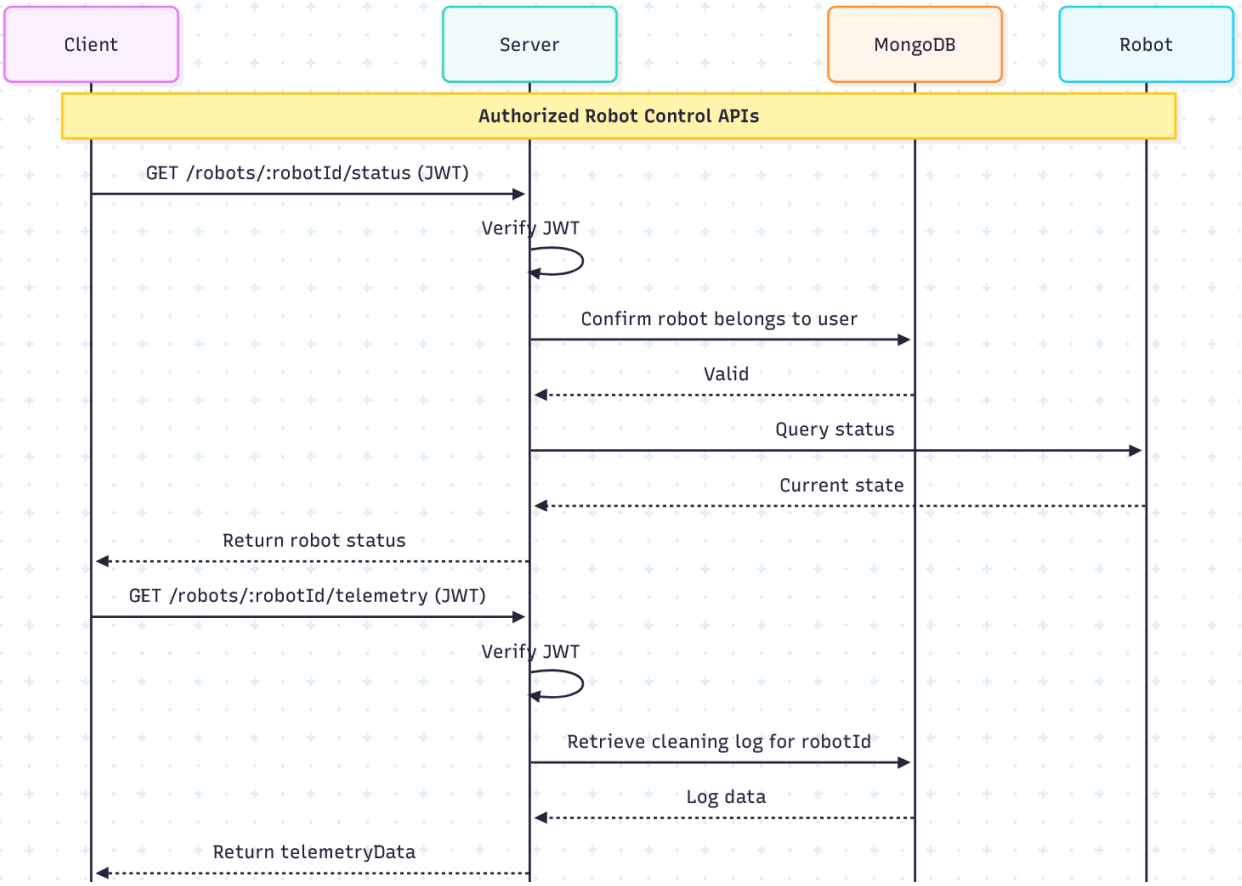

Client
Server
MongoDB
Robot
Public pages (no auth required)
GET /about
Return About page
GET /signin
Return Sign-In Page


Client
Server
MongoDB
Robot
User signup flow
POST /createuser (email, password)
Check if email exists
Result
alt
[Email available]
Hash password
Insert new user
Insert success
User created (201)
[Email exists]
Error (409 Conflict)


Client
Server
MongoDB
Robot
Login flow
POST /login (email, password)
Find user by email
Return user + hash
Verify password
alt
[Valid credentials]
Generate JWT + Refresh token
Tokens returned
[Invalid credentials]
Error (401 Unauthorized)


Client
Server
MongoDB
Robot
Access protected resource
GET /protected (JWT in Authorization header)
Verify JWT signature & expiration
alt
[JWT valid]
Return protected resource
[JWT expired]
POST /refresh (refresh token)
Validate refresh token
Valid
Generate new JWT
Return new JWT


Client
Server
MongoDB
Robot
Robot management
POST /createrobot (JWT, robot details)
Verify JWT
Insert robot with ownerId=userId
Insert success (robotId=r456)
Robot added (201, robotId)
DELETE /robots/:robotId (JWT)
Verify JWT
Delete robot by ID & ownerId
Deletion success
Robot deleted (200)


Client
Server
MongoDB
Robot
Authorized Robot Control APIs
GET /robots/:robotId/status (JWT)
Verify JWT
Confirm robot belongs to user
Valid
Query status
Current state
Return robot status
GET /robots/:robotId/telemetry (JWT)
Verify JWT
Retrieve cleaning log for robotId
Log data
Return telemetryData