

sdmay26-44



Culinary Command

INDUSTRY REVIEW PANEL PRESENTATION

THE TEAM



Matayas Durr
Software Engineering

Frontend Developer
& Client Liaison



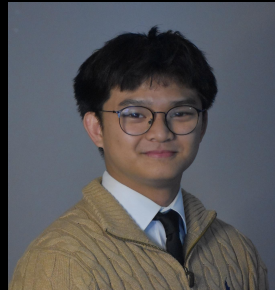
Wyatt Hunter
Software Engineering

Backend &
Architecture Lead



Kevin Tran
Software Engineering

Backend Lead /
Security &
Infrastructure



Anthony Phan
Software Engineering

Frontend & UI/UX
Developer



Ryan Rockey
Software Engineering

Frontend & UI/UX
Developer

PROJECT ORIGIN

Culinary Command began with a real client need.



Paul Durr, co-owner of
Prairie Hospitality Group,
Operates two restaurants in
Grinnell, Iowa.



Needed a better way to assign, track, and communicate daily back-of-house tasks

THE PROBLEM

Restaurants still rely on:

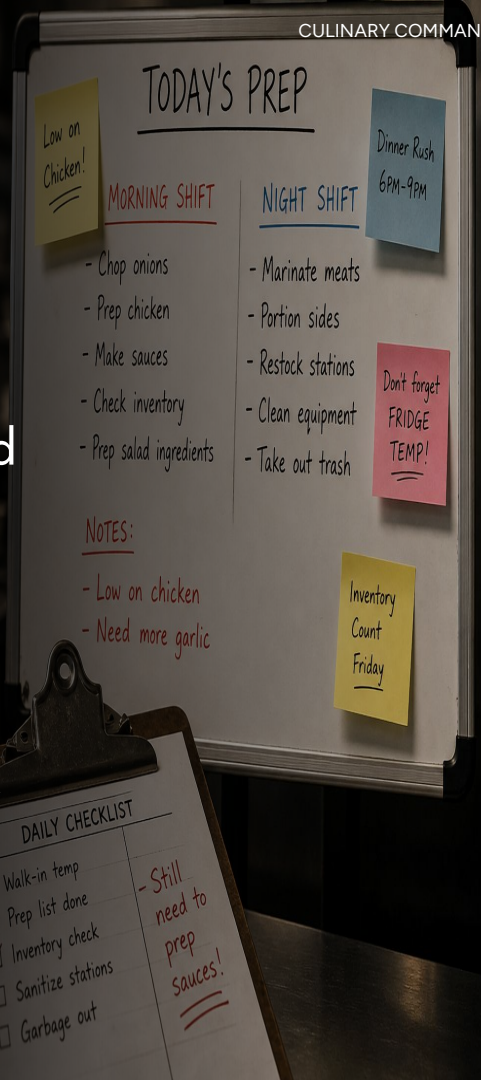
- Verbal communication
- Handwritten prep lists
- Inconsistent workflows



No centralized system for managing back of house tasks.

This leads to:

- Missed or duplicated tasks
- Confusion between shifts
- No accountability or tracking



OUR SOLUTION:

Culinary Command centralizes daily kitchen operations into one role-based web application

How it Works:

- Managers build reusable task lists**
Morning prep, closing, cleaning, inventory checks
- Tasks are assigned by shift or employee**
Employees see only the work they are responsible for
- Progress is tracked throughout the day**
Managers can view pending, in-progress, and completed work
- Inventory supports prep decisions**
Par, count, and prep amounts help determine what needs to be made



Task Management

Templates, task lists, bulk assignment, live status tracking



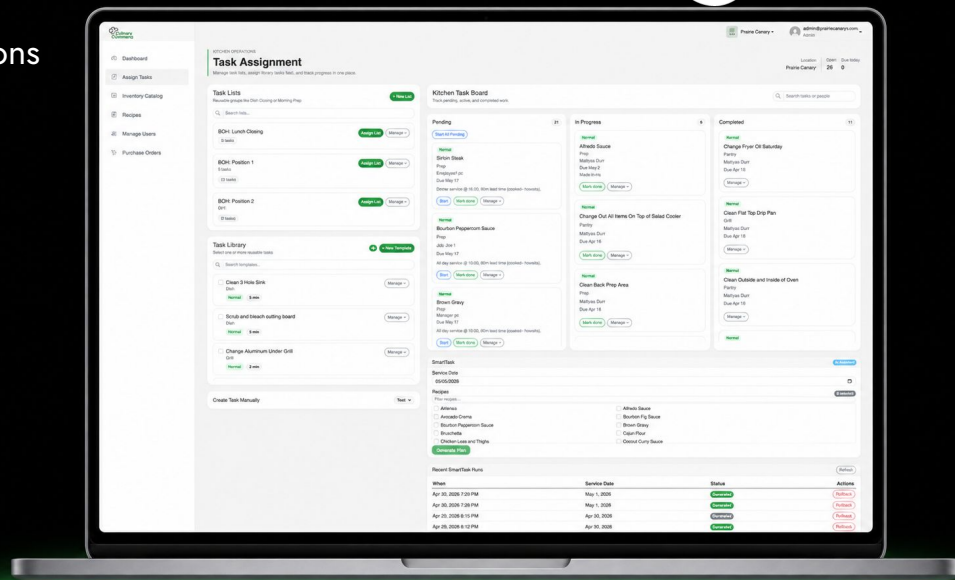
Inventory Management

Par levels, counts, prep amounts, catalog organization



Employee Management

Role-based access, location assignment, personal task views



The system replaces scattered paper lists with a shared workflow for managers and employees.

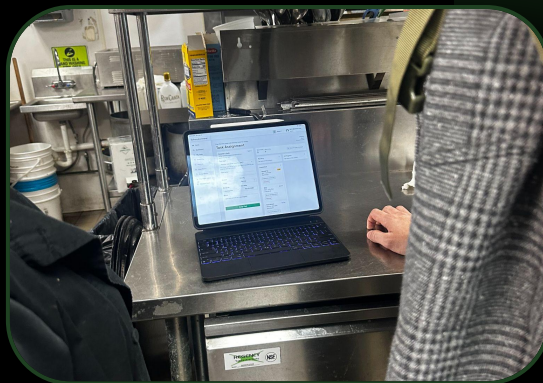
LIVE DEMO & ITERATION

Two in-person demos at Prairie Canary in Grinnell, Iowa



Demo 1

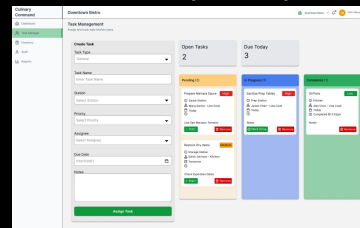
- Early task assignment workflow
- Validated the core concept in a real kitchen
- Used the original Task Assignment Page



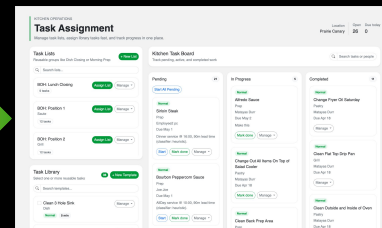
Demo 2

- Demonstrated a more refined system
- Confirmed the product fit the restaurant workflow
- Feedback led to small improvements
 - Employees should be able to view completed tasks again
 - Daily tasks lists appear automatically at set times

Demo 1 (Before)



Current (Refined)



The live demos showed that Culinary Command works in a real kitchen environment and helped guide improvements.

Demo 1 (Before)

Culinary Command | Downtown Bistro | Downtown Bistro | John Manager

Task Management
Assign and track daily kitchen tasks

Create Task

Task Type: General

Task Name:

Station: Select Station

Priority: Select Priority

Assignee: Select Assignee

Due Date: 01/01/2001

Notes:

Assign Task

Open Tasks 2

Pending (2)

- Prep: Maltina Sauce Assign
- Sauce Station
- Maria Garcia - Line Cook
- Today
- Use San Marzano Tomatoes Assign
- Restock Dry Items Station
- Storage Station
- David Johnson - Kitchen
- Tannerwood
- Check Expiration Dates Assign

Due Today 3

In Progress (1)

- Sanitize Prep Tables Assign
- Prep Station
- James Chen - Line Cook
- Today
- Notes
- Mark Done Remove

Completed (1)

- Oil Pans Low
- Kitchen
- Alex Cook - Line Cook
- Today
- Completed @ 3:30pm
- Notes
- Remove



Current (Refined)

KITCHEN OPERATIONS

Task Assignment

Manage task lists, assign library tasks fast, and track progress in one place.

Location: Prairie Canary | Open: 26 | Due Today: 0

Task Lists
Reusable groups like Dish Closing or Morning Prep + New List

Search lists...

- BCH: Lunch Closing Assign List Manage
- BCH: Position 1 Assign List Manage
- BCH: Position 2 Assign List Manage

Task Library
Select one or more reusable tasks + New Template

Search templates...

- Clean 3 Hole Sink Manage
- Dish Normal 5 min

Kitchen Task Board
Track pending, active, and completed work Search tasks or people

Pending 21 Start All Pending

- Normal Start Mark done Manage
- Sirloin Steak
- Prep
- Employed: pc
- Due May 1
- Dinner service @ 18:00, 90m lead time (classifier: heuristic)

In Progress 5

- Normal Mark done Manage
- Alfredo Sauce
- Prep
- Malaysia Durr
- Due May 2
- Malaysia Durr
- Due Apr 18

Completed 11

- Normal Manage
- Change Fryer Oil Saturday
- Grill
- Malaysia Durr
- Due Apr 18
- Clean Flat Top Drip Pan
- Grill
- Malaysia Durr
- Due Apr 18
- Clean Outside and Inside of Oven
- Prep
- Malaysia Durr
- Due Apr 18

Our Approach

1 HOW WE WORKED

Process & Methodology

● Agile sprints → Kanban

Shifted mid-project when schedule constraints made fixed sprints unsustainable

● Two in-person client demos

Iterated based on real kitchen feedback

● Client-driven requirements

Prairie Hospitality Group's needs shaped the feature set

2 HOW WE BUILT

Tech Stack & Architecture

● Blazor Server + .NET9

Modern open-source framework using C# (User Interface)

● Lightsail → EC2 ASG migration

Outgrew original infrastructure and upgraded mid-project

● Modular feature architecture

Inventory, Recipe, SmartTask implemented as isolated modules

3 HOW WE SHIPPED

DevOps & CI/CD

● GitHub Actions pipeline

Automated code checks, unit tests, and end-to-end (E2E) tests on every push

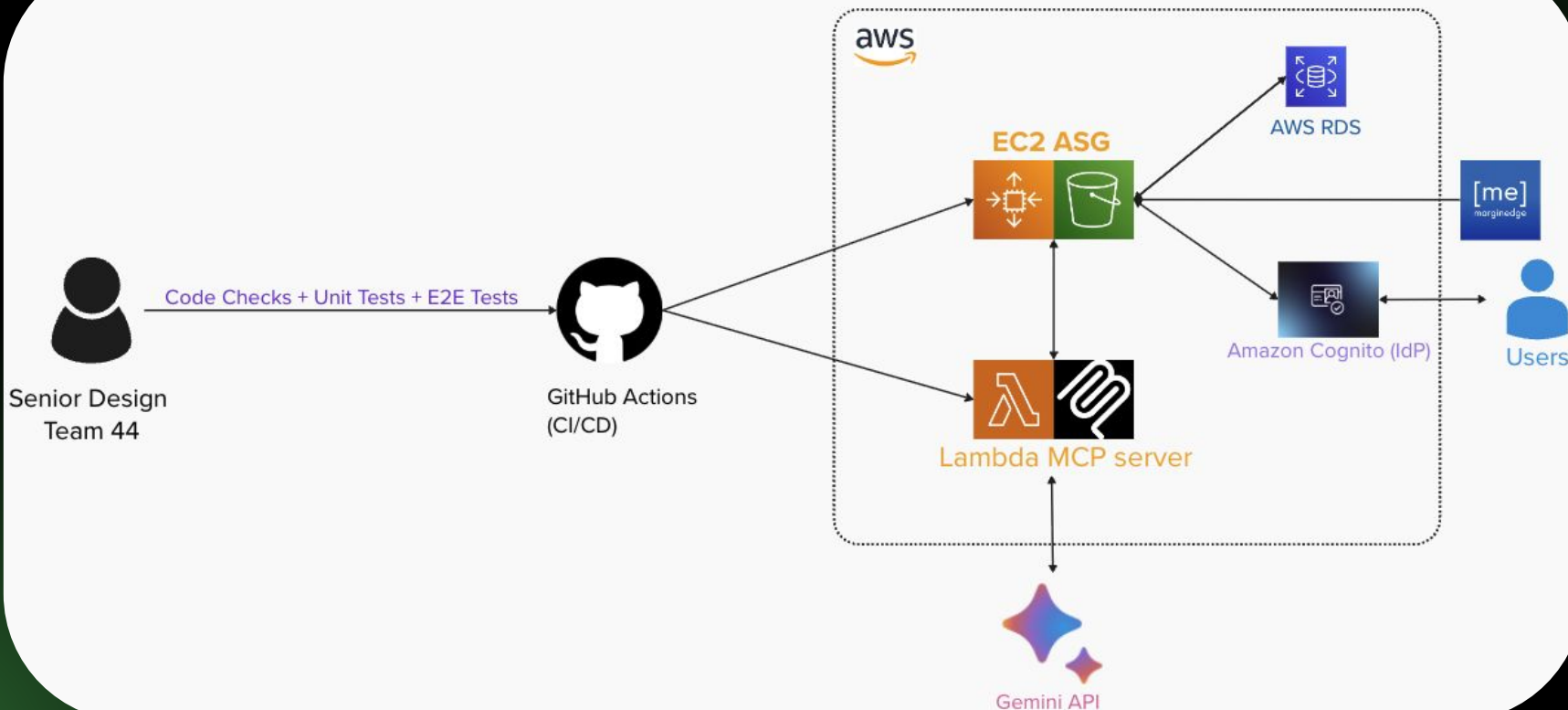
● Layered testing strategy

Unit → Integration → E2E → Load Testing

● Jira project board

Shared view of capacity, everyone knew what was in flight and what was next

System Architecture Overview





GitHub Actions

Delivering value to our workflow

346

TOTAL MINUTES

CI/CD time invested

241

TOTAL JOB RUNS

automated executions

55s

AVG JOB RUN TIME

fast feedback loops

3s

AVG QUEUE TIME

nearly instant start

8%

JOB FAILURE RATE

26 minutes lost to failed jobs
92% of jobs succeed on first run

WHY IT MATTERS

- 320 minutes of productive CI time
- 241 manual tasks eliminated
- 55s average feedback loops enable rapid iteration
- 3s queue time means developers never wait to start

EC2 Auto Scaling Architecture

Fault Resistant

Instances span across 3 availability zones

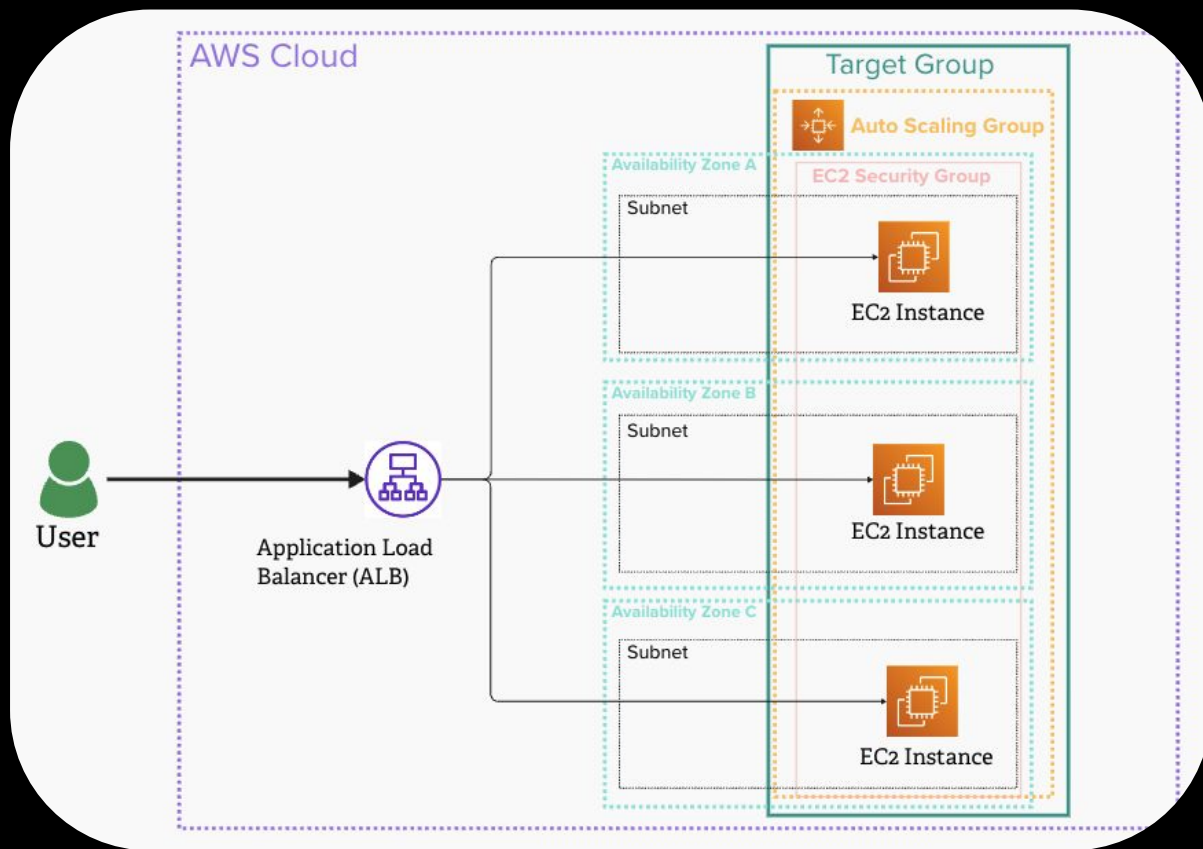
Elastic Capacity

Auto Scaling Group adds or removes instances based on demand

Resilient By Design

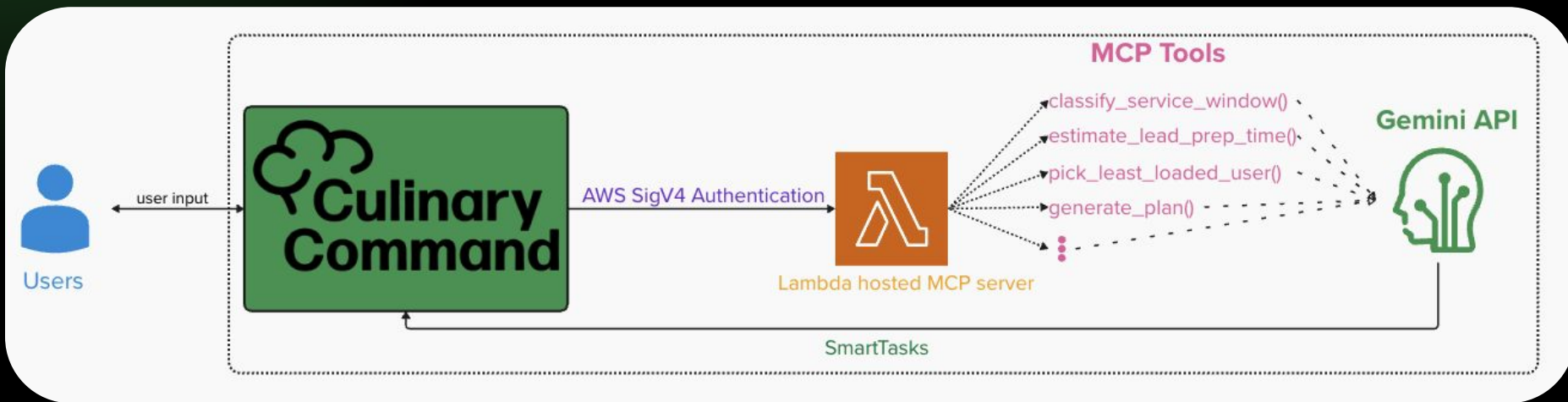
Failed instances are detected and replaced without any manual intervention

Application load balancer spreads traffic evenly across all instances

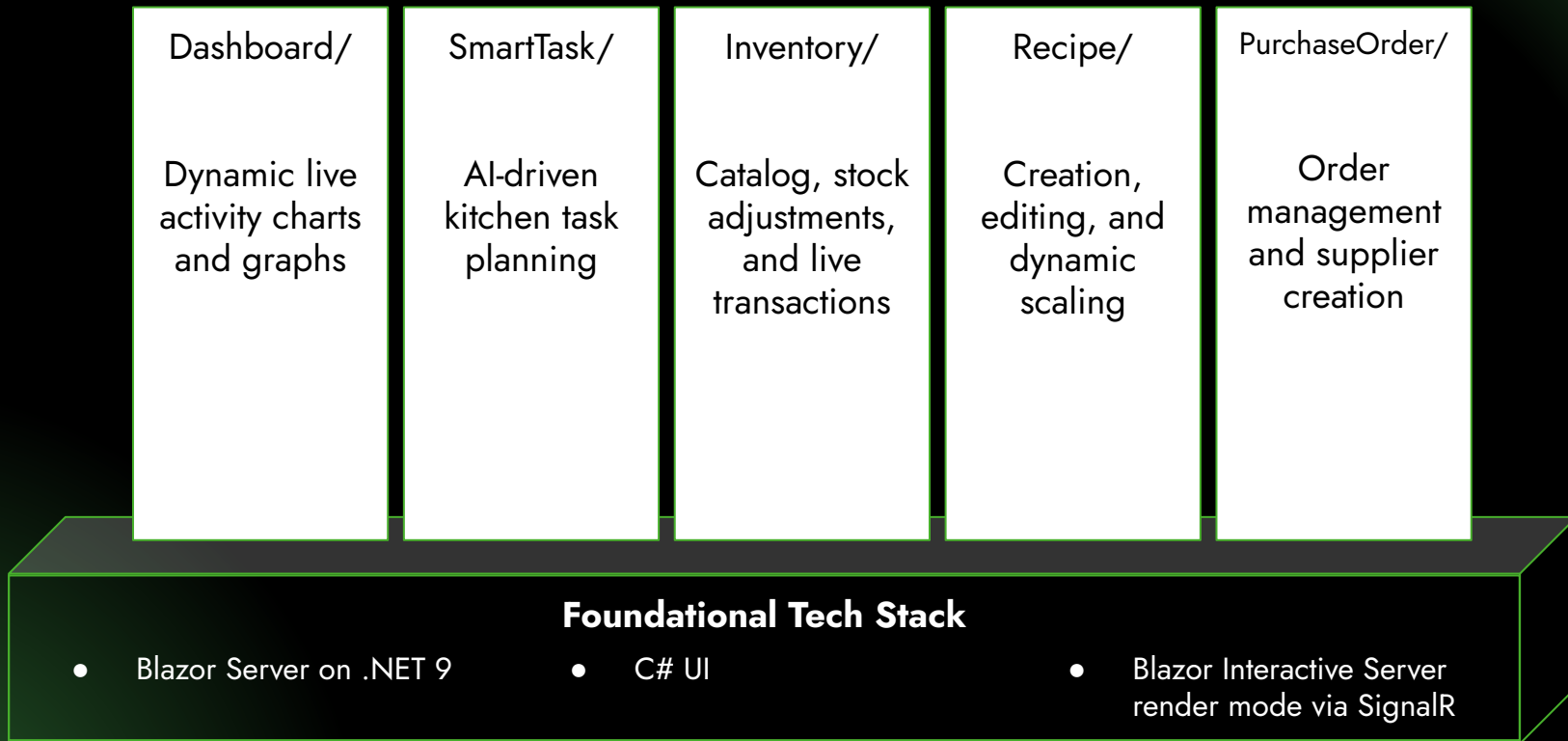


Model Context Protocol (MCP) Server Architecture

An MCP server allows us to connect our Culinary Command application to an LLM.

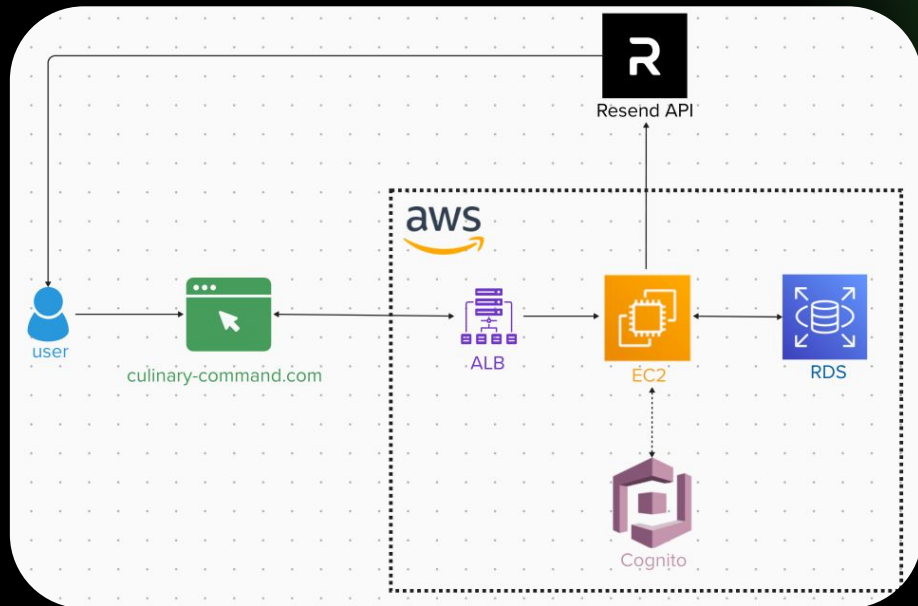


Tech Stack & Modular Architecture



Backend Architecture

- Blazor Server + ASP.NET
 - Client (browser) connects to server via SignalR
- Client:
 - Sends user events (clicks, inputs)
 - Receives UI updates from server



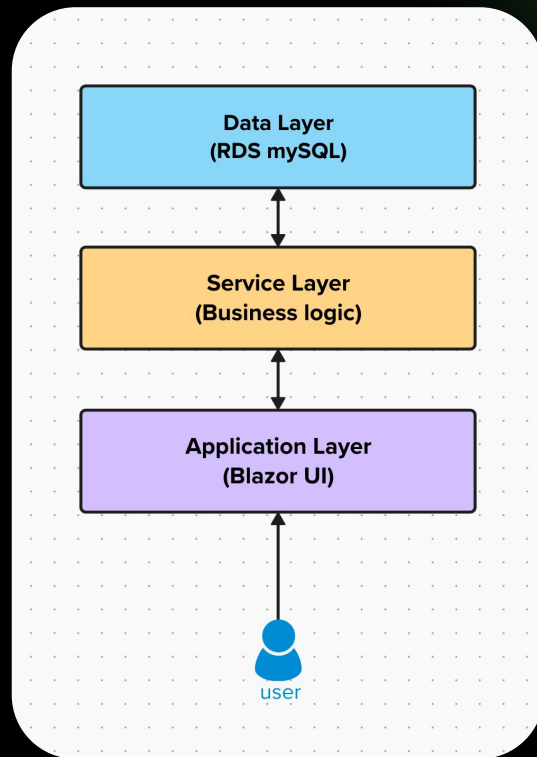
Lightsail vs. EC2

- Blazor is more backend heavy
- Memory constraints with Lightsail
- Multi-region zones ensure highest uptime



Service Layer

- Layer between UI and external systems
- Example for User & Auth:
 - UserService,
CognitoProvisioningService
- Interface-driven design
 - TaskService implements ITaskService
- Dependency injection
 - Loose coupling



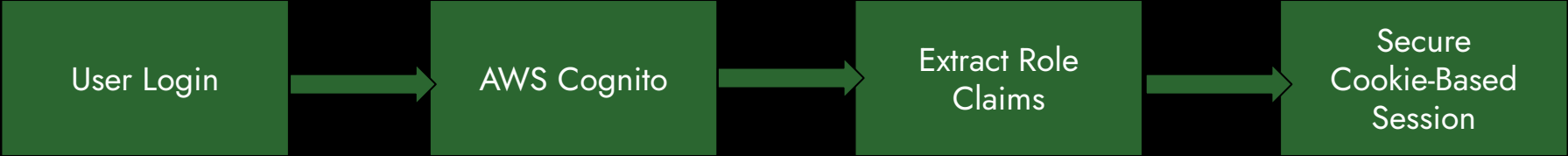
Integrations

- Authentication and user provisioning
 - AWS Cognito
- Email system for onboarding
 - Resend API
- Database migrations and access
 - .NET EF Core

Resend

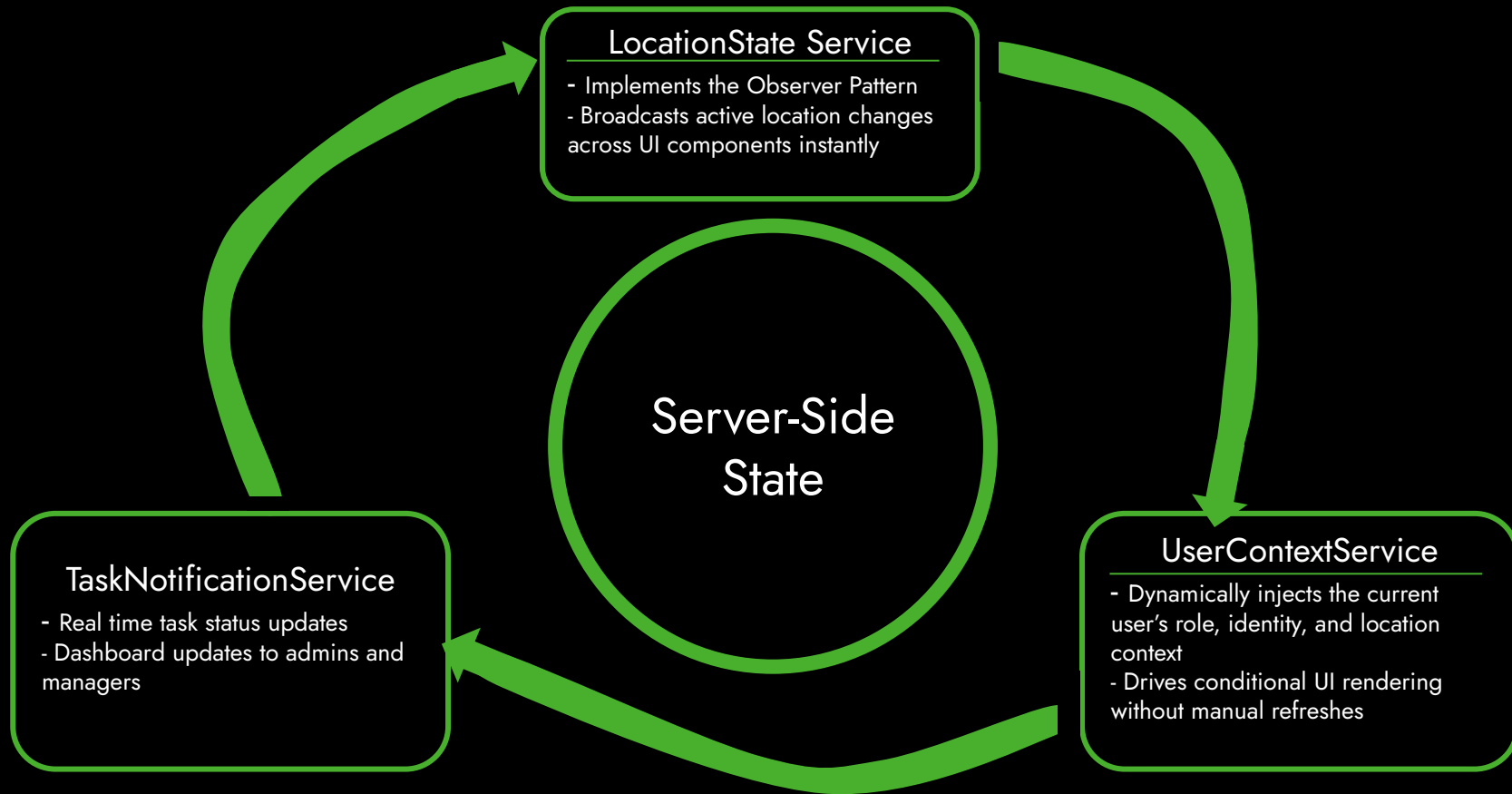


Frontend Role Based Access

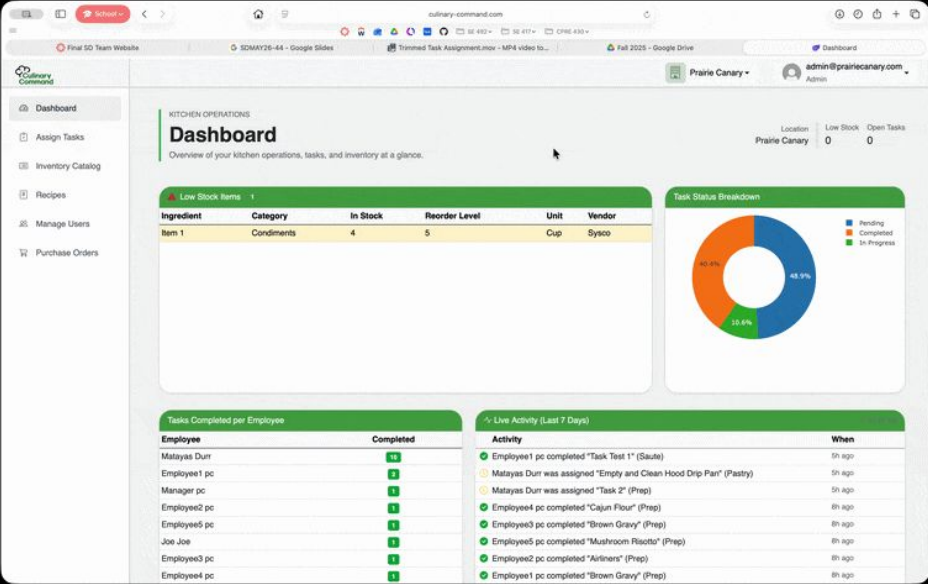


Admin	Manager	Employee
- Complete system control - Global company configurations - User lifecycle management	- Dashboard analytics access - Manual & AI task management - Inventory, Recipes & Purchase orders	- Strictly locked down interface - Rendered "My Tasks" only - Minimal views

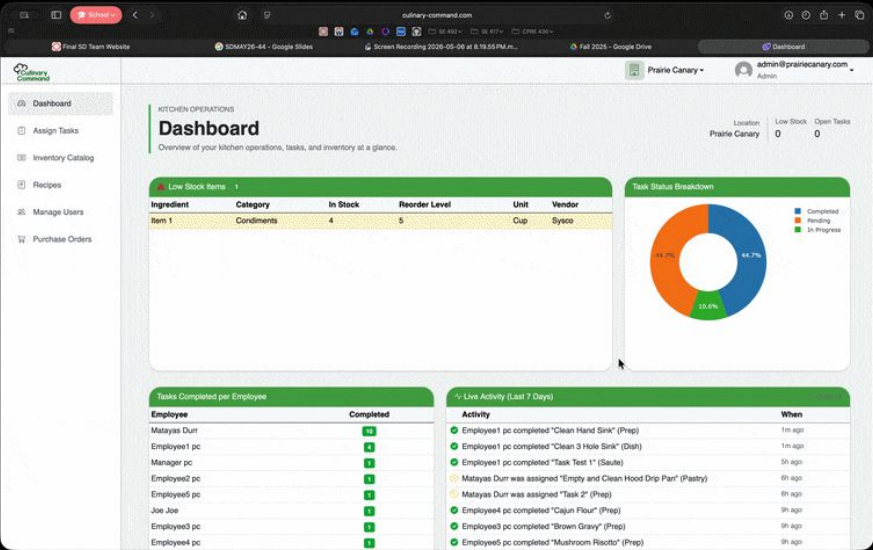
Frontend State Management & Real Time Sync



Frontend Demos



LocationStateService



TaskNotificationService

Frontend Views

Culinary Command

Dashboard

Assign Tasks

Inventory Catalog

Recipes

Manage Users

Purchase Orders

KITCHEN OPERATIONS

Dashboard

Overview of your kitchen operations, tasks, and inventory at a glance.

Location

Prairie Canary

Low Stock

0

Open Tasks

0

Low Stock Items

1

Ingredient	Category	In Stock	Reorder Level	Unit	Vendor
Item 1	Condiments	4	5	Cup	Sysco

Task Status Breakdown

38.3%

31.1%

10.6%

Completed

Pending

In Progress

Tasks Completed per Employee

Employee	Completed
Matayas Durr	10
Employee1 pc	7
Manager pc	5
Employee2 pc	3
Employee3 pc	2
Joe Joe	1
Employee3 pc	1
Employee4 pc	1
Wyatt Hunter	1

Live Activity (Last 7 Days)

Activity

When

Employee1 pc completed "Pull Burners & Spray off in Sink" (Saute)

just now

Employee1 pc completed "Clean Splash Guard Behind Burners" (Saute)

just now

Employee1 pc completed "Clean Sauce Rail in Front of Burner" (Saute)

just now

Employee1 pc completed "Clean Hand Sink" (Prep)

1m ago

Employee1 pc completed "Clean 3 Hole Sink" (Dish)

2m ago

Employee1 pc completed "Task Test 1" (Saute)

5h ago

Matayas Durr was assigned "Empty and Clean Hood Drip Pan" (Pastry)

6h ago

Matayas Durr was assigned "Task 2" (Prep)

6h ago

Employee4 pc completed "Caun Flour" (Prep)

9h ago

Admin/Manager View

Culinary Command

My Tasks

KITCHEN OPERATIONS

My Tasks

Track your assigned work, review recipe prep, and close out tasks quickly.

Show Completed

Location

Prairie Canary

Open

10

Recipe Prep

3

ASSIGNED TASKS

Today's Task List

Review assigned work, expand general task notes, and complete prep with recipe context where needed.

Date and Put Away Steam well Pans

General

Apr 18

Mark Done

Refill Cooking Liquids

General

Apr 18

Mark Done

Scrub and bleach cutting board

General

Apr 18

Mark Done

Take Out and Replace Trash Bags

General

Apr 18

Mark Done

Wipe/Clean Cooler

General

Apr 18

Mark Done

Wrap all pans in Sauté Station

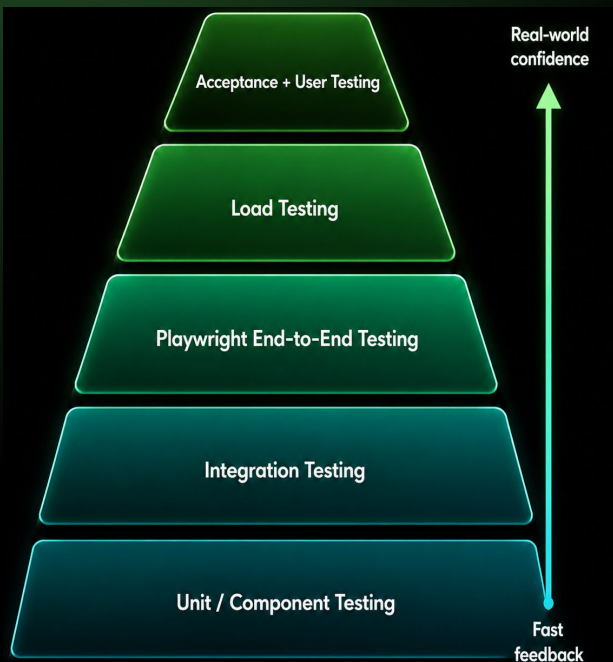
General

Apr 18

Mark Done

21

Testing Strategy: Layered Validation



Goal: Validate critical workflows across the UI, backend, database, and authentication system.



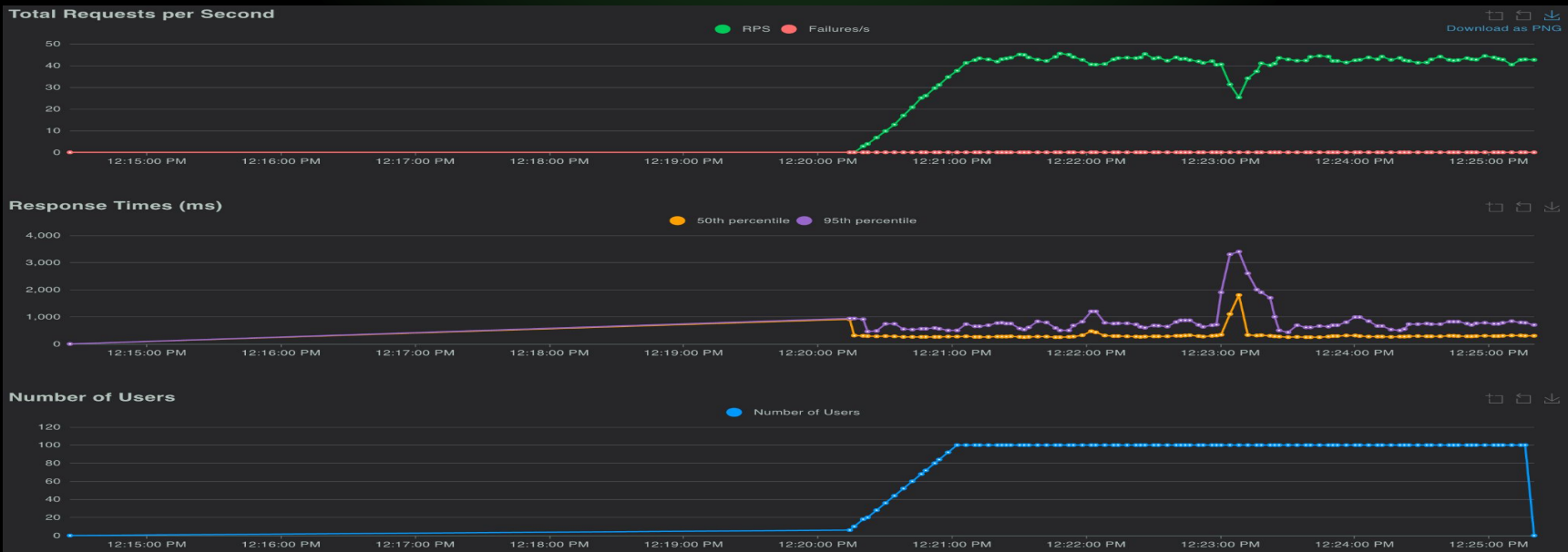
- **Unit/component testing** - validate models, DTOs, and business rules
- **Integration tests** - verify services, database behavior, and role-based access control
- **Playwright E2E testing** - simulate real Admin, Manager, and Employee workflows in the browser
- **Locust load testing** - measure responsiveness under simulated traffic
- **Acceptance / User testing** - validate workflows with client-focused scenarios

Automated Workflow Testing

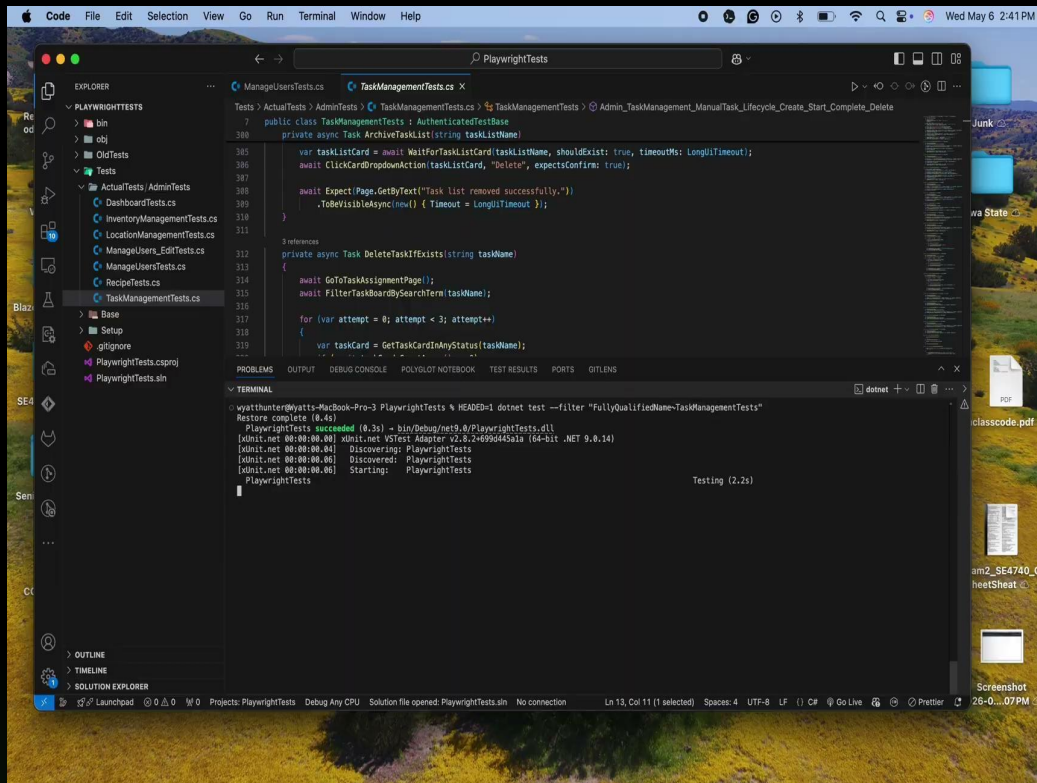
<u>Area</u>	<u>Validated</u>
Authentication	Cognito redirect + authenticated dashboard access
User Management	Invites, CRUD operations, validation, role assignment
Task Management	CRUD, assign, update status, verify board changes
Inventory	CRUD, search/filter, update stock, low-stock detection
Recipes/Ingredients	CRUD, connect ingredients, produce validation
Location/Tenant Views	Tenant-specific data and location based access
Role Views	Admin, Manager, Employee workflow visibility

Load Testing

- Used Locust to simulate HTTP traffic against Culinary Command
- Measured request volume, active users, and response-time percentiles
- Tracked average/median behavior and ranked 95th percentile worst-case response time
- Established a baseline for infrastructure capacity
- Future work: repeat load tests as features and real restaurant data increase



PlayWright Example



Takeaways

- Adaptability matters more than a perfect plan
- Working with real cloud infrastructure gave us an experience much closer to industry software development than a typical classroom assignment
- Learned the importance of end-to-end testing as the application scaled
- User feedback helped prioritize which features actually delivered value instead of just being technically interesting

Q&A