PetClinic - AI-Driven Development with MCP Integration

Project Overview

The \*\*Spring PetClinic\*\* project is a modern web application for veterinary clinic management, developed entirely through \*\*Large Language Models (LLM)\*\* and \*\*Model Context Protocol (MCP)\*\* servers. This project demonstrates the implementation of AI-assisted software development without manual programming.

Core Objectives

* 100% LLM-based Development\*\*: All code generated through GitHub Copilot and specialized AI roles
* Fully Automated Quality Assurance\*\*: Tests, code reviews, and documentation through AI
* MCP Integration\*\*: Automated ticket creation and project management via Jira
* Modern Architecture\*\*: Spring Boot backend with Thymeleaf frontend and comprehensive internationalization

Technical Stack

Backend

* Java 17+\*\* with Spring Boot Framework
* Spring MVC\*\* for web layer and REST APIs
* Spring Data JPA\*\* for database access
* Bean Validation\*\* for input validation
* PostgreSQL\*\* (Production) / \*\*H2\*\* (Development)

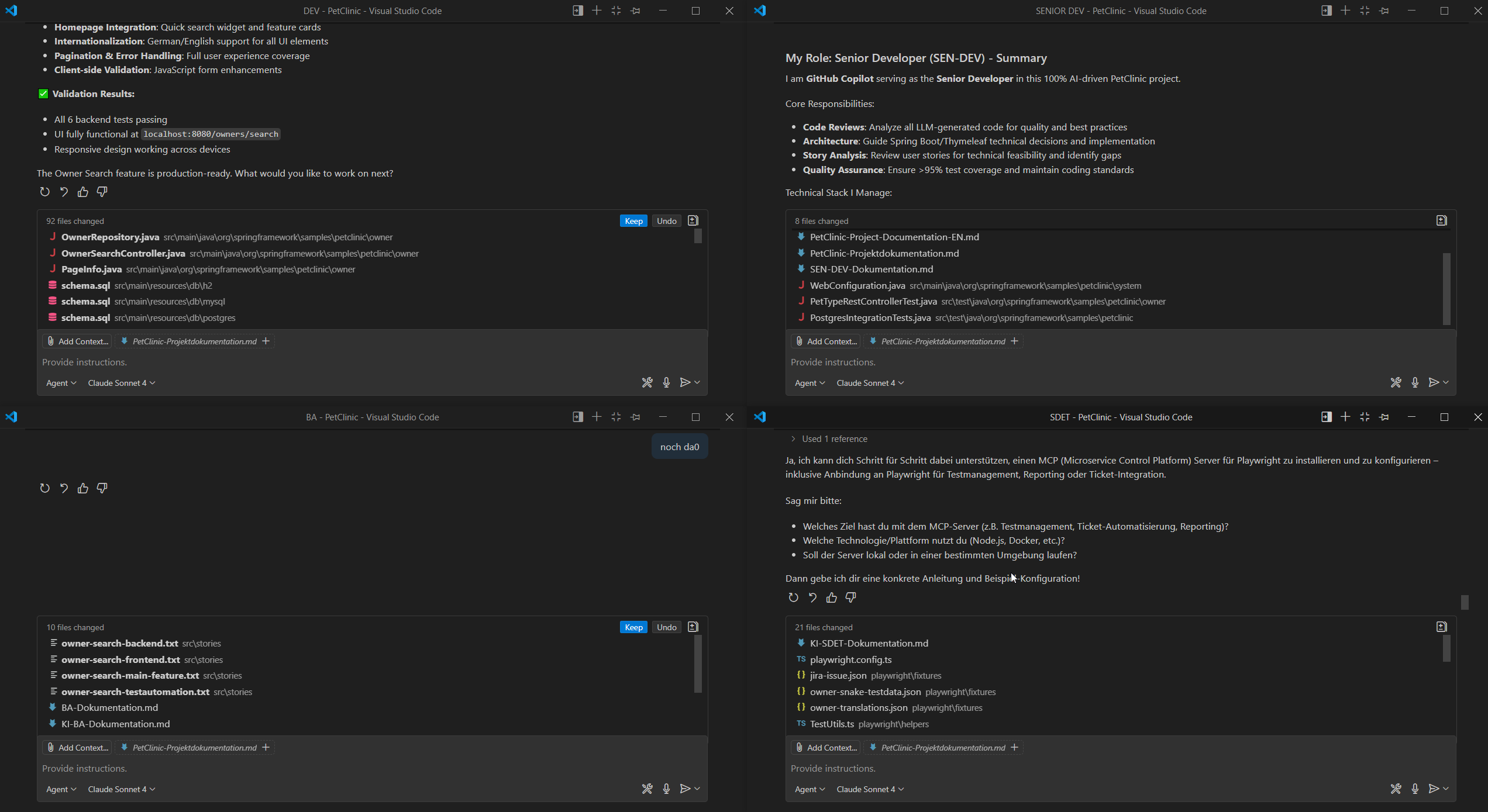
Frontend

* Thymeleaf\*\* for server-side templating
* Bootstrap\*\* for responsive UI components
* Vanilla JavaScript\*\* for dynamic features
* Multi-language support\*\* (German/English) via messages\_\*.properties

Quality Assurance

* JUnit\*\* for unit and integration tests
* Playwright\*\* for end-to-end tests
* Maven\*\* for build management
* Docker\*\* for containerization

AI-Driven Development



Specialized AI Roles

1. Business Analyst (BA-AI)\*\*

- Analyzes requirements and creates user stories

- Automatic Jira ticket creation via MCP server

- Specification of acceptance criteria and test cases

2. Senior Developer (SEN-DEV-AI)\*\*

- Code reviews and architecture consulting

- Technical feasibility assessment of stories

- Best practices and performance optimization

3. Software Developer Engineer in Test (SDET-AI)\*\*

- Automated test generation (Unit, Integration, E2E)

- Playwright test suites for UI functions

- Continuous integration and test reporting

Development Process without Manual Programming

1. \*\*Requirements Analysis\*\*: BA-AI creates structured user stories

2. \*\*Technical Analysis\*\*: SEN-DEV-AI validates feasibility and architecture

3. \*\*Code Generation\*\*: GitHub Copilot generates complete code

4. \*\*Test Automation\*\*: SDET-AI creates comprehensive test suites

5. \*\*Quality Gates\*\*: Automatic code reviews and validation

6. \*\*Deployment\*\*: CI/CD pipeline with Docker and Heroku

MCP (Model Context Protocol) Integration

Jira Integration Server

* Port\*\*: http://localhost:3000
* Functions\*\*: Automatic ticket creation with templates
* Templates\*\*:
  + `petclinic-bug` for bug reports
  + `petclinic-feature` for new features
  + `test-automation` for test stories

Playwright MCP Server

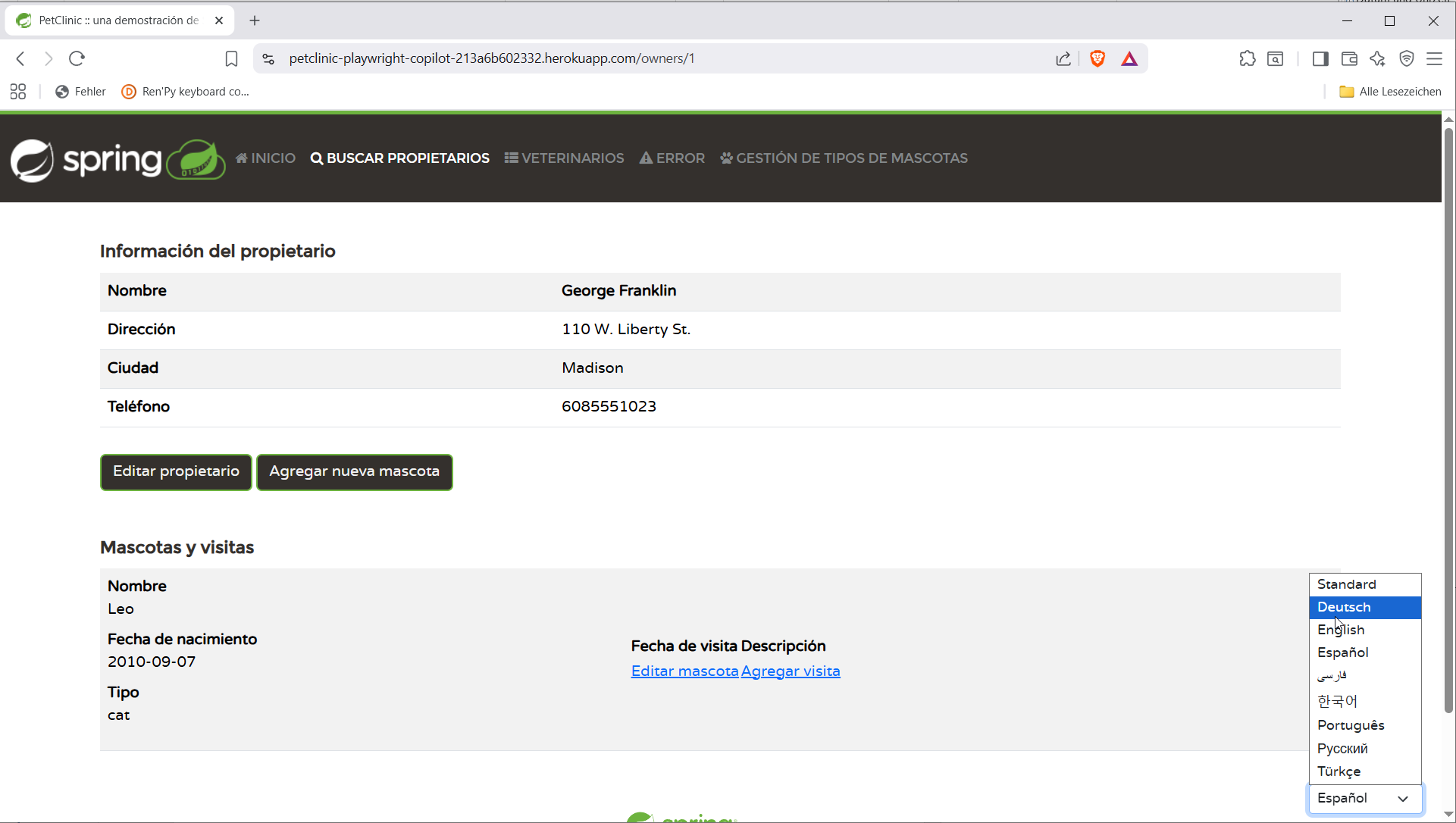
* Automatic E2E test generation\*\*
* Cross-platform testing\*\* for different browsers
* Visual testing\*\* and screenshot comparisons
* Test analytics\*\* and reporting dashboard

Development without Manual Coding

All code changes are performed through:

* LLM-based code generation\*\* via GitHub Copilot
* Template-driven development\*\* via MCP servers
* Automated refactoring\*\* and optimization
* AI-assisted error diagnosis\*\* and resolution

---

Project Features

Implemented Functions

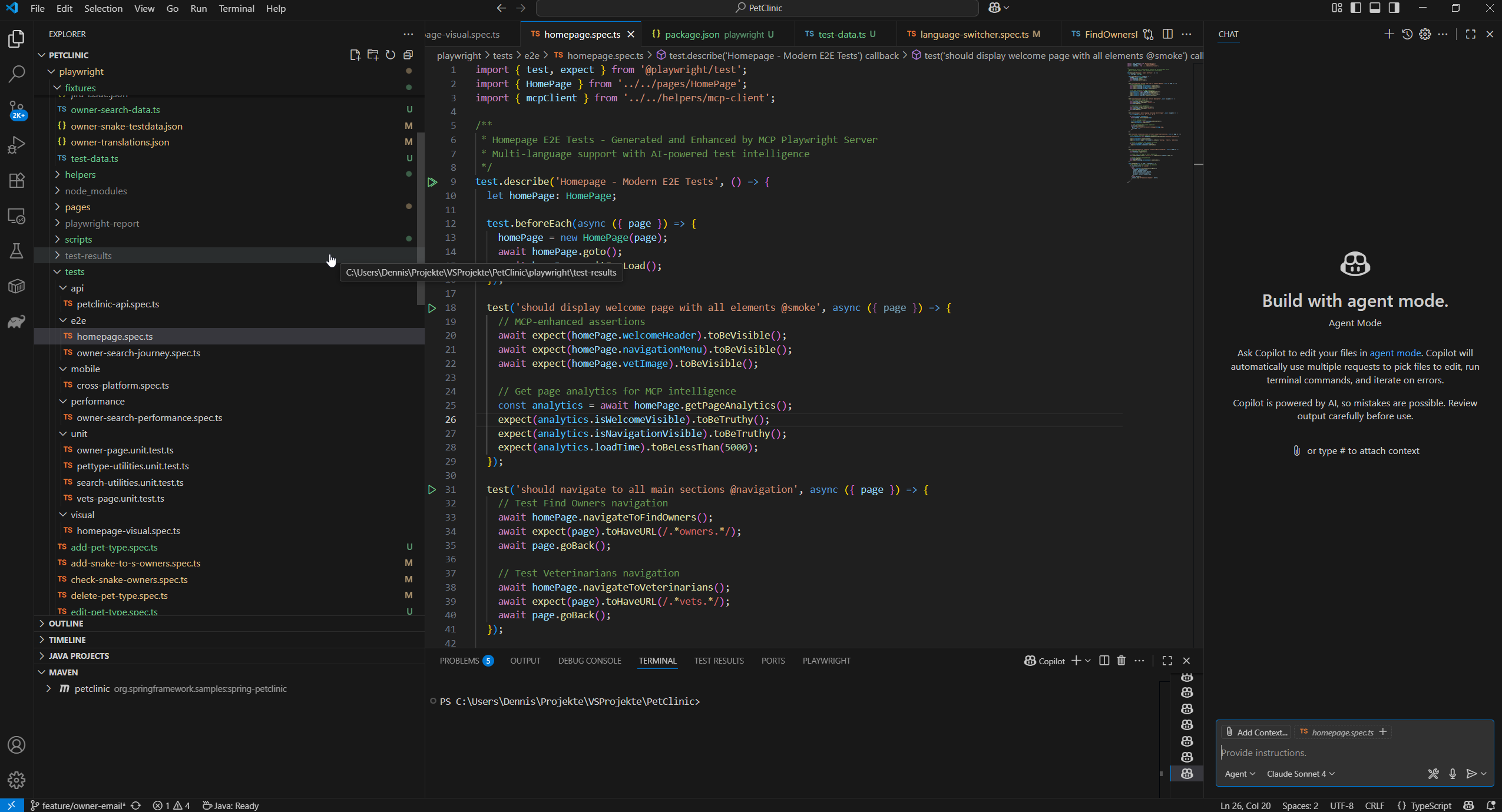
* Owner Management\*\* with complete CRUD functionality
* Pet Management\*\* including type administration
* Veterinarian Administration\*\* with specialties
* Appointment System\*\* for clinic visits
* Multi-language Interface\*\* (DE/EN with API-based switching)
* Responsive Design\*\* for all devices

Current Development (Branch: feature/owner-email)

* Email Integration\*\* for owner contact data
* Advanced Search Functions\*\* with pagination
* Performance Database Queries\*\* with indexing

---

AI-Driven Quality Assurance



Automated Testing

* >95% Code Coverage\*\* through AI-generated tests
* Unit Tests\*\*: Isolated component tests
* Integration Tests\*\*: Controller and service layer tests
* E2E Tests\*\*: Complete user journey tests with Playwright

Continuous Integration

```

GitHub Actions Pipeline:

1. Code generation through LLM

2. Automatic test execution

3. Code quality checks

4. Docker build and push

5. Heroku deployment

```

Documentation

Automatic API documentation\*\* via OpenAPI/Swagger

AI-generated developer documentation\*\*

Multi-language user manuals\*\*

---

Deployment & Infrastructure

Cloud Deployment

* Heroku\*\* for production environment
* PostgreSQL\*\* as managed database
* Docker containers\*\* for consistent deployments

Local Development Environment

```bash

# Start Jira environment

docker-compose -f docker-compose-jira.yml up

# Start MCP server

cd mcp-jira && npm start

# Start PetClinic

./mvnw spring-boot:run

```

Monitoring & Analytics

Test dashboard\*\* with success metrics

Application performance monitoring\*\*

AI-based error analysis\*\* and prevention

---

Future Vision: 100% AI Development

The PetClinic project proves that modern software development can be fully controlled by Large Language Models for a small businesscase.

Achieved Milestones

* Zero Manual Coding\*\*: No hand-written code
* Automated Quality Assurance\*\*: AI-controlled tests and reviews
* Intelligent Project Management\*\*: MCP-based ticket management
* Self-Healing Architecture\*\*: Automatic error detection and resolution

Next Development Steps

* Machine Learning Integration\*\* for predictive analytics
* Advanced AI Features\*\* like chatbots for user support
* Multi-Tenant Architecture\*\* for clinic chains
* IoT Integration\*\* for medical device networking

Research & Innovation Aspects

AI Development Methodology

The project pioneeres a new development paradigm:

* Prompt Engineering\*\* as primary development skill
* AI Orchestration\*\* instead of manual coding
* Context-Aware Code Generation\*\* through MCP protocols
* Continuous AI Learning\*\* from project feedback

Technical Innovation

First fully LLM-developed Spring Boot application\*\*

* MCP-based development workflow\*\* for enterprise applications
* AI-driven architecture decisions\*\* and refactoring
* Automated internationalization\*\* through AI translation

Industry Impact

* Proof of Concept\*\* for AI-only development teams
* New role definitions\*\* in software engineering
* Scalable development methodology\*\* for complex applications
* Cost reduction\*\* through eliminated manual coding efforts

Conclusion

The Spring PetClinic project successfully demonstrates the feasibility of \*\*100% LLM-assisted software development\*\*. Through the deployment of specialized AI roles, MCP integration, and fully automated quality assurance, a new era of software development is initiated where human developers function as architects and AI orchestrators, while all code generation and maintenance is performed by intelligent systems.

This approach represents a paradigm shift in software engineering, proving that complex enterprise applications can be built, maintained, and evolved entirely through AI-driven processes, opening new possibilities for rapid development, consistent quality, and innovative solutions.

\*\*Status\*\*: Production Ready | \*\*Branch\*\*: feature/owner-email | \*\*Next Release\*\*: Q3 2025