GitHub Copilot Workshop

From Zero to Hero: Mastering AI-Powered Development

Welcome to the comprehensive GitHub Copilot workshop for the Demo Inventory Microservice project! This workshop will guide you through mastering GitHub Copilot for full-stack development using Clean Architecture, .NET 9, and React 19.

🎯 Workshop Objectives

By the end of this workshop, you will:

- **Master GitHub Copilot basics** in VS Code and GitHub Chat
- **Build features end-to-end** using AI assistance
- **Write comprehensive tests** with Copilot's help
- **Debug and fix issues** efficiently using AI
- **Follow best practices** for AI-assisted development
- **Validate AI-generated code** for quality and security

📚 Workshop Structure

📖 Part 1: Extended Training Session

Duration: 3.5 hours (with breaks)
Comprehensive foundation and advanced knowledge for AI-assisted development
Learning Outcome: Master the fundamentals of GitHub Copilot configuration, prompt engineering, and best practices for professional AI-assisted development

🚀 Getting Started Guide

Duration: 45 minutes
Comprehensive setup and introduction to GitHub Copilot

- Installation and configuration
- Basic and advanced prompting techniques
- Understanding project context and workspace optimization
- Your first AI-generated code
- Hands-on practice with simple examples

⚙️ Copilot Instruction Configuration

Duration: 30 minutes
Deep dive into configuration for optimal performance

- Project-specific instructions
- Context optimization for Clean Architecture
- Custom configuration for .NET 9 and React 19
- Team collaboration settings
- Environment-specific configurations

🧠 Model Context Protocol (MCP) Guide

Duration: 40 minutes
Advanced context management with MCP for enhanced AI development

- Understanding MCP and its benefits for development
- Step-by-step MCP setup in VS Code
- Using MCP for context-aware code generation
- Practical MCP patterns and use cases
- Advanced debugging and analysis with MCP
- Performance optimization and security enhancement

🏆 Best Practices Guide

Duration: 45 minutes
Comprehensive practices for effective AI-assisted development

- Advanced prompt engineering techniques
- Context management strategies
- Code quality maintenance
- Security considerations
- Performance optimization with AI
- Team collaboration patterns

✅ Code Verification Guide

Duration: 30 minutes
Advanced validation techniques for AI-generated code

- Quality assessment techniques
- Security review checklist
- Architecture compliance validation
- Testing verification methods
- Code review best practices with AI
- Static analysis integration

🔧 Troubleshooting Guide

Duration: 20 minutes
Comprehensive problem-solving approach

- Setup and configuration problems
- Copilot not working as expected
- Code generation issues
- Integration problems
- Performance and optimization issues

☕ Break 1

Duration: 10 minutes
Refreshment break and networking opportunity to discuss initial AI experiences and setup challenges

🛠️ Part 2: Core Hands-On Exercises

Duration: 4 hours (with breaks)
Practical coding exercises to master AI-assisted development
Learning Outcome: Gain hands-on experience building complete features using Clean Architecture with AI assistance, from backend APIs to frontend components and comprehensive testing

1. Backend Development Exercise

Duration: 75 minutes
Build complete features using Clean Architecture

- Create domain entities with complex business rules
- Implement application services and DTOs
- Add repository patterns and EF Core configuration
- Build API controllers with advanced validation
- Performance optimization techniques
- Security implementation patterns

2. Frontend Development Exercise

Duration: 75 minutes
Create sophisticated React components with TypeScript

- Generate TypeScript interfaces from backend DTOs
- Build complex forms with validation and state management
- Implement API integration with error handling
- Add loading states and user experience enhancements
- Responsive design patterns
- Accessibility considerations

☕ **Break 2**

Duration: 15 minutes
Refreshment break and progress discussion - share insights from backend/frontend development exercises

3. Testing Exercise

Duration: 60 minutes
Comprehensive testing strategy with AI assistance

- Generate unit tests with xUnit and NSubstitute
- Create integration tests for API endpoints
- Build E2E tests with Cypress
- Performance testing strategies
- Security testing approaches

4. Bug Fixing Exercise

Duration: 45 minutes
Advanced debugging and problem resolution

- Analyze complex error messages and stack traces
- Generate debugging solutions for architectural issues
- Fix performance problems and memory leaks
- Handle concurrency and async issues
- Validate fixes with comprehensive tests

☕ Break 3

Duration: 10 minutes
Quick break before advanced topics - reflect on testing and debugging techniques learned

🚀 Part 3: Advanced Topics and Scenarios

Duration: 4 hours (with breaks)
Advanced AI-assisted development techniques and real-world scenarios
Learning Outcome: Master advanced AI techniques for complex scenarios including refactoring legacy code, implementing security best practices, optimizing performance, and managing DevOps processes

5. Refactoring Exercise

Duration: 45 minutes
Advanced code quality improvement with AI assistance

- Identify complex code smells and refactoring opportunities
- Apply SOLID principles and design patterns
- Optimize performance and memory usage
- Maintain Clean Architecture boundaries
- Legacy code modernization techniques

6. Security-Focused Development

Duration: 60 minutes
AI-assisted secure coding practices

- Security vulnerability detection and prevention
- Input validation and sanitization
- Authentication and authorization implementation
- Secure configuration management
- Security testing and code reviews

7. Performance Optimization Exercise

Duration: 50 minutes
AI-assisted performance tuning and optimization

- Database query optimization
- Memory management and garbage collection
- Async/await best practices
- Caching strategies implementation
- Load testing and profiling

☕ Break 4

Duration: 15 minutes
Refreshment break and technical discussions - exchange experiences with refactoring, security, and performance optimization

8. DevOps and Deployment Exercise

Duration: 60 minutes
AI-assisted DevOps practices and deployment

- Docker containerization optimization
- CI/CD pipeline enhancement
- Infrastructure as Code with AI
- Monitoring and logging setup
- Cloud deployment strategies

9. Team Collaboration Scenarios

Duration: 45 minutes
Advanced team development with AI

- Code review automation with AI
- Pair programming with Copilot
- Knowledge sharing and documentation
- Onboarding new team members
- Best practices for AI in team environments

10. Architecture Evolution Exercise

Duration: 45 minutes
AI-assisted architectural decision making

- Microservices patterns and implementation
- Event-driven architecture design
- API design and versioning strategies
- System scalability planning
- Technology stack evolution

🏆 Part 4: Capstone Project and Review

Duration: 1 hour
Apply everything learned in a comprehensive project
Learning Outcome: Demonstrate mastery by building a complete feature from scratch using all learned AI-assisted development techniques, and receive feedback for continued improvement

11. Capstone Project

Duration: 45 minutes
Build a complete feature from scratch

- Design and implement a new inventory tracking feature
- Apply all learned techniques and best practices
- Use AI for entire development lifecycle
- Demonstrate mastery of AI-assisted development

🎓 Workshop Review and Reflection

Duration: 15 minutes
Review achievements and plan next steps
What you'll learn: How to assess your AI development skills, identify areas for improvement, and create a personal learning plan for continued AI-assisted development growth

- Knowledge assessment and certification
- Feedback and improvement suggestions
- Next steps for continued learning
- Resources for ongoing development

🎓 Learning Path

12.5-Hour Comprehensive Track

Total Duration: 12.5 hours with breaks
Complete mastery of AI-assisted development

Part 1: Extended Training Session (3.5 hours)

1. Getting Started Guide (45 min)
2. Copilot Instruction Configuration (30 min)
3. Model Context Protocol (MCP) Guide (40 min)
4. Best Practices Guide (45 min)
5. Code Verification Guide (30 min)
6. Troubleshooting Guide (20 min)
7. Break 1 (10 min)

Part 2: Core Hands-On Exercises (4 hours)

1. Backend Development Exercise (75 min)
2. Frontend Development Exercise (75 min)
3. Break 2 (15 min)
4. Testing Exercise (60 min)
5. Bug Fixing Exercise (45 min)
6. Break 3 (10 min)

Part 3: Advanced Topics and Scenarios (4 hours)

1. Refactoring Exercise (45 min)
2. Security-Focused Development (60 min)
3. Performance Optimization Exercise (50 min)
4. Break 4 (15 min)
5. DevOps and Deployment Exercise (60 min)
6. Team Collaboration Scenarios (45 min)
7. Architecture Evolution Exercise (45 min)

Part 4: Capstone Project and Review (1 hour)

1. Capstone Project (45 min)
2. Workshop Review and Reflection (15 min)

Flexible Learning Options

Half-Day Track (6.5 hours)

For Teams with Limited Time
Learn core AI-assisted development skills for immediate productivity gains

- Part 1: Extended Training Session (2.5 hours - including essential MCP setup)
- Part 2: Core Hands-On Exercises (4 hours)

- **What you'll learn**: Essential Copilot usage, prompt engineering basics, code generation fundamentals, and hands-on practice with backend/frontend development
- **Outcome**: Ability to use GitHub Copilot effectively for daily development tasks

Advanced Specialist Track (8 hours)

For Senior Developers and Architects
Master advanced AI techniques for complex architectural and performance challenges

- Part 1: Extended Training Session (1.5 hours - skip basics)
- Part 2: Core Hands-On Exercises (2.5 hours - accelerated)
- Part 3: Advanced Topics and Scenarios (4 hours - full focus)
- **What you'll learn**: Advanced prompt engineering, architectural decision-making with AI, security-focused development, performance optimization, and complex refactoring techniques
- **Outcome**: Expertise in using AI for architectural decisions, security implementation, and system optimization

Team Leader Track (10 hours)

For Technical Leads and Managers
Develop AI adoption strategies and team leadership skills for AI-assisted development

- All sections with additional focus on:
- Team collaboration scenarios
- Code review processes
- Adoption strategies
- Training team members
- **What you'll learn**: Team AI adoption strategies, code review best practices for AI-generated code, mentoring techniques, and organizational change management for AI tools
- **Outcome**: Ability to lead AI adoption initiatives and train development teams effectively

🛠️ Prerequisites

Note: These prerequisites ensure you can focus on learning AI-assisted development techniques rather than struggling with basic setup. The workshop will build upon this foundation to teach advanced AI integration patterns.

Required Tools

- **VS Code** with recommended extensions (latest version)
- **GitHub Copilot** subscription and access (Business or Individual)
- **.NET 9 SDK** for backend development
- **Node.js 20+** for frontend development
- **Docker Desktop** for containerization and deployment
- **PostgreSQL** for database (or Docker alternative)
- **Git** for version control
- **Postman** or similar API testing tool

Required Knowledge

- **Intermediate C#** and .NET fundamentals
- **Intermediate TypeScript/JavaScript** and React concepts
- **Good understanding** of REST APIs and HTTP protocols
- **Familiarity with Git** and GitHub workflows
- **Basic understanding** of Clean Architecture concepts

- **Experience with** database concepts and SQL

Recommended Knowledge (for advanced sections)

- **Docker** and containerization concepts
- **CI/CD** pipeline basics
- **Cloud platforms** (Azure, AWS, or GCP)
- **Performance testing** and optimization
- **Security** best practices

Advanced Prerequisites (for specialist tracks)

- **Software architecture** patterns and principles
- **DevOps** practices and tools
- **Team leadership** or technical mentoring experience

Setup Verification

Before starting, ensure your environment is ready by running these commands. This verification process will also familiarize you with the project structure and build process that you'll be working with throughout the workshop:

```
# Verify .NET version
dotnet --version  # Should be 9.0+

# Verify Node.js version
node --version    # Should be 20.0+

# Verify Docker
docker --version

# Clone the workshop repository (if not already done)
git clone https://github.com/zeabix-cloud-native/demo-inventory-microservice.git
cd demo-inventory-microservice

# Verify project builds - this confirms your development environment is properly configure
dotnet build
cd frontend && npm install && npm run build
```

🎯 Success Metrics

Track your progress with these comprehensive checkpoints:

Foundation Knowledge Checkpoints

- ☐ Can configure GitHub Copilot for optimal performance in any environment
- ☐ Can write effective prompts for complex code generation scenarios
- ☐ Can generate complete features using Clean Architecture principles
- ☐ Can create comprehensive tests with AI assistance
- ☐ Can debug complex issues efficiently using Copilot
- ☐ Can validate and verify AI-generated code for quality and security

Advanced Development Checkpoints

- ☐ Built multiple complete API endpoints with advanced validation
- ☐ Created sophisticated React components with TypeScript interfaces

- [ ] Generated comprehensive test suites (unit, integration, E2E)
- [ ] Fixed complex bugs using AI-assisted debugging techniques
- [ ] Refactored legacy code following SOLID principles
- [ ] Implemented security best practices with AI assistance

Specialization Checkpoints

- [ ] Optimized application performance using AI-guided techniques
- [ ] Designed and implemented secure coding patterns
- [ ] Created and configured CI/CD pipelines with AI assistance
- [ ] Led team collaboration sessions using AI tools
- [ ] Made architectural decisions with AI support
- [ ] Delivered a complete capstone project from conception to deployment

Team Leadership Checkpoints

- [ ] Can train other developers in AI-assisted development
- [ ] Can establish team standards for AI tool usage
- [ ] Can review and validate AI-generated code at enterprise scale
- [ ] Can integrate AI tools into existing development workflows
- [ ] Can measure and report on AI adoption benefits

🤝 Getting Help

During the Workshop

- **Ask questions** in VS Code using `@workspace` commands
- **Reference existing code** for patterns and examples
- **Use the troubleshooting guide** for common issues
- **Share your screen** during collaborative sessions

Resources and Support

- Project Documentation
- Architecture Guide
- Development Setup
- GitHub Copilot Documentation
- VS Code Copilot Extension

🚀 Ready to Start?

1. **Verify your setup** using the commands above
2. **Begin with Part 1: Training Session**
   - Start with the Getting Started Guide
   - Configure Copilot Instructions for this project
   - Read through Best Practices and Code Verification
3. **Choose your learning track** based on your experience level
4. **Move to Part 2: Hands-On Exercises** to practice your skills
5. **Experiment, practice, and have fun!**

Remember: The goal is not just to use Copilot, but to become proficient at AI-assisted development while maintaining code quality, security, and architectural integrity.

Let's build something amazing together! 🚀