Building with Claude Code

Office Betting Demo

Claude Code

From idea to deployed app in one session

> **Important Notice**
>
> The company has not yet released an AI policy permitting the use of Claude Code for work purposes.
>
> Please use personal time and resources only for any experimentation with this tool.

Today's agenda:

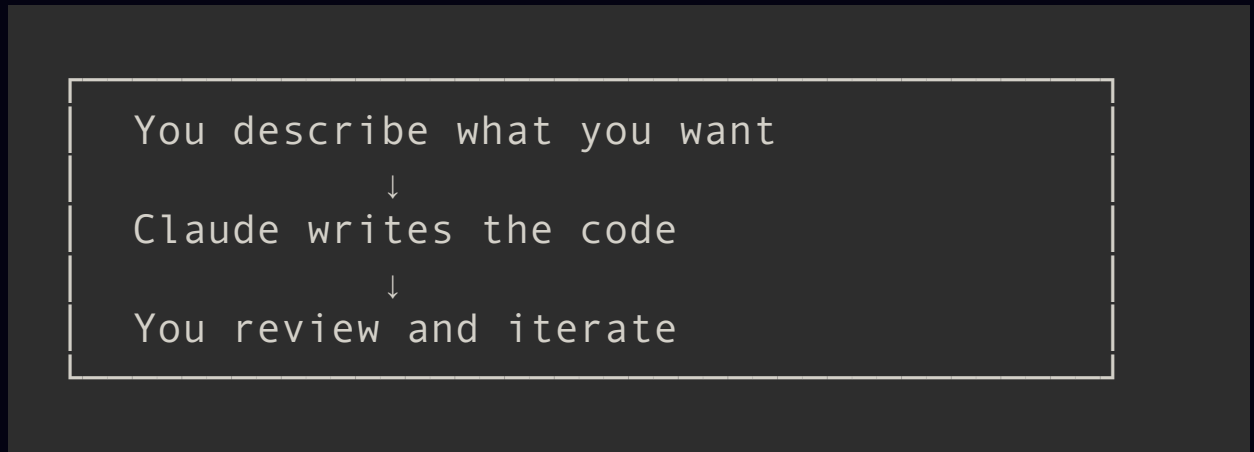
1. What is Claude Code?
2. Live demo: Office Betting Platform
3. How it was built
4. Running the project

What is Claude Code?

A CLI tool that brings Claude AI directly into your terminal

Key features:

- Reads and writes files in your codebase
- Runs shell commands
- Understands your entire project context
- Works with any language or framework



Getting Started Resources

▓▓ We have two companion tutorials!

1. Getting Started with Claude Code

`getting-started-claude-code.md`

- Subscribing to Claude Pro/Max
- Installation on Mac and Windows
- Authentication setup

2. Claude Code Tutorial

`claude-code-tutorial.md`

- Commands and workflows
- Planning mode, permissions
- Context management, MCP servers
- Tips and best practices

Live Demo: Office Betting

What we built in one session

A pari-mutuel betting platform with:

- FastAPI backend (Python 3.12)
- React + TypeScript frontend
- SQLite database
- JWT authentication
- Docker Compose deployment

One prompt to start:

> "Build a complete web application for an office play-money betting platform using pari-mutuel betting"

Project Structure

```
office-betting/
├── backend/
│   ├── src/mirustech/betting/
│   │   ├── main.py              # FastAPI app
│   │   ├── models/              # SQLAlchemy models
│   │   ├── routers/             # API endpoints
│   │   └── services/            # Business logic
│   └── Dockerfile
├── frontend/
│   ├── src/
│   │   ├── pages/               # React pages
│   │   └── components/          # UI components
│   └── Dockerfile
└── docker-compose.yml
```

All generated by Claude Code!

Key Features Built

- Pari-mutuel betting logic

```
def calculate_payout(wager, winning_outcome,
total_pool):
    user_weighted = wager.amount * wager.weight
    return (user_weighted /
winning_outcome.weighted_total)
           * total_pool
```

- Early bet bonus (1.2x weight)

```
def calculate_weight(bet, wager_time):
    total_window = bet.close_time - bet.created_at
    halfway = bet.created_at + (total_window / 2)
    return 1.2 if wager_time < halfway else 1.0
```

Running Locally

Quick start (without Docker)

```
# Backend
cd backend
uv venv && uv pip install -e .
source .venv/bin/activate
python -m mirustech.betting.seed  # Load demo data
uvicorn mirustech.betting.main:app --port 8000

# Frontend (new terminal)
cd frontend
npm install && npm run dev
```

Open in browser

Docker Deployment

One command startup

```
docker-compose up --build
```

What it does

1. Builds backend image (Python + FastAPI)
2. Builds frontend image (Node + Nginx)
3. Creates SQLite volume for persistence
4. Seeds demo data automatically

Access the app

```
http://localhost:3000
```

Takeaways

▓▓ Claude Code enables:

- Rapid prototyping - Full stack app in hours, not days
- Best practices - Follows your coding standards
- Iterative development - Fix issues conversationally
- Learning - See how experienced code is structured

Next Steps

▓▓▓ Want to try Claude Code yourself?

Step 1: Review the tutorials

```
presenterm getting-started-claude-code.md
presenterm claude-code-tutorial.md
```

Step 2: Get a Claude Pro or Max subscription

Visit: https://claude.ai

Step 3: Install and start experimenting!

> Remember: Personal time and resources only
> until company AI policy is released.