

MVP Tech Stack for Summer 2026 Internship

Target Companies

Microsoft, Wise, Twilio, Ericsson

Core Stack

Backend

Python + FastAPI

- Modern, async, type-safe Python framework
- Auto-generated API documentation (Swagger UI)
- Fast development with production-grade performance
- Shows backend depth and modern Python skills

Frontend

Next.js + Tailwind CSS

- React-based framework with built-in routing
- Server-side rendering capabilities
- Tailwind for rapid, clean UI development
- Industry-standard, highly hireable skill

Database

PostgreSQL

- Hosted on Supabase or Railway
- Demonstrates relational database modeling
- Production-ready, scalable solution
- Easy integration with auth and storage

Authentication

Supabase Auth or Clerk

- Drop-in authentication solution
 - Social login providers included
 - Production-ready security out of the box
 - Saves development time for core features
-

Deployment Strategy

Backend Deployment

- **Platform:** Railway, Render, or fly.io
- **Container:** Docker
- **Benefits:** Shows containerization skills, production deployment experience

Frontend Deployment

- **Platform:** Vercel
- **Benefits:** One-click deployment, automatic previews, optimized performance

Database Hosting

- **Platform:** Managed PostgreSQL (Supabase or Railway)
- **Benefits:** Zero database operations overhead, automatic backups

Development Tools

Version Control

- Git + GitHub
- Clean commit history
- Professional README with architecture diagram

Environment Management

- python-dotenv for backend
- Environment variables for frontend
- Proper .env file management

Project Structure

- requirements.txt for Python dependencies
- package.json for Node dependencies
- Clear separation of frontend/backend

Optional Enhancements

(Add if time permits)

Performance

- **Redis** for caching
- Shows understanding of performance optimization

Background Processing

- **Celery** for async task queues
- Demonstrates handling of long-running operations

Testing

- **Pytest** for backend tests
- Huge plus for code quality demonstration

Why This Stack Impresses Recruiters

Technical Competence

- Modern, production-grade technologies
- Full-stack capability demonstration
- Cloud deployment experience

Architecture Understanding

- Clean separation of concerns
- RESTful API design
- Proper authentication implementation

Professional Practices

- Containerization with Docker
- Environment configuration
- Version control best practices

Interview Talking Points

- Technology choice reasoning
- Architecture decision trade-offs
- Scaling considerations
- Error handling strategies

Key Success Factors

1. **Working Product** - Functionality over perfection
2. **Clean Code** - Readable, well-organized codebase
3. **Great README** - Clear documentation with setup instructions
4. **Live Demo** - Deployed and accessible application
5. **Explanation Ready** - Understand every technology choice

Next Steps

1. Define your product idea
2. Set up development environment
3. Create GitHub repository
4. Build MVP iteratively
5. Deploy early and often
6. Document thoroughly
7. Prepare for technical discussions

Architecture Pattern



Frontend (Next.js)

↓ HTTP/REST API

Backend (FastAPI)

↓ SQL queries

Database (PostgreSQL)

Note: Next.js API routes won't be used - frontend calls FastAPI backend directly for clean separation of concerns.