1. **Codebase Download**

Download all project files

1. **Required Environment Variables**

**Create a .env file in the root directory**:

# Required for AI Dashboard Generation

OPENAI\_API\_KEY=your\_openai\_api\_key\_here

# Database Configuration (Production)

DATABASE\_URL=postgresql://username:password@host:port/database

PGHOST=your\_postgres\_host

PGPORT=5432

PGUSER=your\_username

PGPASSWORD=your\_password

PGDATABASE=acgme\_analytics

# Azure B2C Configuration (Production)

AZURE\_CLIENT\_ID=your\_azure\_client\_id

AZURE\_TENANT=your\_tenant\_name

AZURE\_POLICY=B2C\_1\_signin

# Application Configuration

NODE\_ENV=production

PORT=5000

1. **Installation Commands**

# Install dependencies

npm install

# For development with in-memory database

npm run dev

# For production with PostgreSQL

npm run build

npm start

1. **File Structure Overview**

acgme-analytics/

├── client/ # React frontend

│ ├── src/

│ │ ├── components/ # UI components

│ │ ├── pages/ # Page components

│ │ ├── hooks/ # Custom React hooks

│ │ └── lib/ # Utilities and configuration

│ └── index.html # HTML template

├── server/ # Express backend

│ ├── services/ # Business logic services

│ ├── index.ts # Server entry point

│ ├── routes.ts # API endpoints

│ └── storage.ts # Data access layer

├── shared/ # Shared TypeScript types

│ └── schema.ts # Database schema

├── attached\_assets/ # User-provided assets

├── package.json # Dependencies and scripts

├── vite.config.ts # Build configuration

├── tailwind.config.ts # Styling configuration

└── .env # Environment variables

**5. Critical Integration Points**

See ACGME\_Analytics\_Design\_Document.md for detailed integration instructions for:

* Power BI data connections
* Azure B2C authentication
* Row-level security implementation
* OpenAI API optimization
* ACGME public data APIs

**6. Deployment Options**

**Replit Deployment**: Click "Deploy" in Replit interface for instant public URL **Manual Deployment**: Use any Node.js hosting service (Vercel, Heroku, AWS, Azure) **Enterprise Deployment**: Configure with your organization's infrastructure

**7. Testing the Application**

1. Start the development server: npm run dev
2. Open [http://localhost:5000](http://localhost:5000/)
3. Login with demo credentials or test authentication
4. Try AI queries like "Show cardiology programs by state"
5. Verify geographic map displays correctly
6. Test dashboard generation and data visualization

**8. Troubleshooting**

**Common Issues**:

* Missing OpenAI API key: Add to .env file
* Port conflicts: Change PORT in .env
* Database connection: Verify DATABASE\_URL format
* Missing dependencies: Run npm install again

**Support**: Refer to the comprehensive design document for detailed technical specifications and integration guidance.