FutureGen Interview Automation System

Setup and Documentation Guide

Shreeraj Mummidivarapu
+91 8143272388

September 17, 2025

Comprehensive guide for setting up, configuring, and running the FutureGen Interview Automation System

---

Contents

1 Introduction

Overview

The FutureGen Interview Automation System is a comprehensive web application designed to streamline the interview process through automated registration, resume analysis, and interview management capabilities.

2 Technology Stack

2.1 Frontend Architecture

- **Framework:** React 18 with modern hooks and functional components
- **UI Components:** Custom CSS with responsive design, Lucide React icons
- **State Management:** React Context API for global state
- **HTTP Client:** Native fetch API with error handling
- **Routing:** React Router for single-page application navigation

2.2 Backend Architecture

- **Framework:** FastAPI (Python) with automatic API documentation
- **Database:** PostgreSQL with JSONB support for flexible data storage
- **ORM:** SQLAlchemy with declarative models
- **Authentication:** JWT (JSON Web Tokens) with secure token handling
- **Migrations:** Alembic for database version control
- **Validation:** Pydantic models for request/response validation

2.3 Development Tools

- **Package Management:** npm for frontend, pip for backend
- **API Testing:** FastAPI Swagger UI with interactive documentation
- **Development Server:** Hot reloading for both frontend and backend
- **Environment Management:** Python virtual environments

3 Prerequisites

System Requirements

Ensure the following software is installed on your system before proceeding:

1. **Node.js** (v16.0.0 or higher)
   - Download from: `https://nodejs.org/`
   - Verify installation: `node --version`
2. **Python** (3.9.0 or higher)
   - Download from: `https://python.org/`
   - Verify installation: `python --version`
3. **PostgreSQL** (13.0 or higher)
   - Download from: `https://postgresql.org/`
   - Ensure `psql` command is available
4. **Git** (for version control)
   - Download from: `https://git-scm.com/`
   - Required for cloning the repository

4 Project Structure

Directory Layout

Understanding the project structure is crucial for navigation and development.

```
futuregen-interview/
          src/                              # Frontend React components
                components/                  # Reusable UI components
                pages/                       # Page-level components
                      AdminDashboard.js        # Admin interface
                      AdminLogin.js            # Admin authentication
                      User.js                  # User interface
                context/                     # React Context providers
                utils/                       # Utility functions
                styles/                      # CSS stylesheets
                App.js                       # Main application component
                index.js                     # Application entry point
          backend/                         # Python FastAPI backend
                models/                      # SQLAlchemy database models
                      user.py                  # User model
                      registration.py          # Registration model
                routes/                      # API route handlers
                      auth.py                  # Authentication routes
                      users.py                 # User management routes
                      registrations.py         # Registration routes
```

```
                schemas/                    # Pydantic validation schemas
                database.py                 # Database configuration
                dependencies.py             # FastAPI dependencies
                main.py                     # FastAPI application entry
    point
        alembic/                          # Database migration files
                versions/                   # Migration version files
                env.py                      # Alembic environment config
                alembic.ini                 # Alembic configuration
        .env.example                      # Environment variables template
        requirements.txt                  # Python dependencies
        package.json                      # Node.js dependencies and scripts
```

Listing 1: Complete Project Structure

5 Setup Instructions

5.1 Backend Setup

5.1.1 Virtual Environment

```
# Create virtual environment
python -m venv venv

# Activate virtual environment
# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate

# Verify activation (should show venv path)
which python
```

Listing 2: Create Python Virtual Environment

5.1.2 Install Dependencies

```
# Install all required packages
pip install -r requirements.txt

# Verify installation
pip list | grep fastapi
pip list | grep sqlalchemy
```

Listing 3: Install Python Packages

5.1.3 Environment Configuration

Create a `.env` file in the project root directory:

```
# Database Configuration
DATABASE_URL=postgresql://username:password@localhost:5432/futuregen_db
DB_HOST=localhost
DB_PORT=5432
```

```
DB_NAME=futuregen_db
DB_USER=your_username
DB_PASSWORD=your_password

# Security Configuration
SECRET_KEY=your-super-secure-secret-key-here-min-32-chars
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=30

# Application Configuration
DEBUG=True
API_V1_STR=/api
PROJECT_NAME="FutureGen Interview Automation"

# CORS Configuration
ALLOWED_HOSTS=["localhost", "127.0.0.1"]
CORS_ORIGINS=["http://localhost:3000", "http://127.0.0.1:3000"]
```

Listing 4: Environment Variables (.env file)

5.1.4 Database Setup

```
# Connect to PostgreSQL as superuser
psql -U postgres

# Create database and user
CREATE DATABASE futuregen_db;
CREATE USER your_username WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE futuregen_db TO your_username;

# Exit PostgreSQL
\q

# Test database connection
psql -U your_username -d futuregen_db -h localhost
```

Listing 5: PostgreSQL Database Setup

5.1.5 Database Migrations

```
# Navigate to project root
cd /path/to/futuregen-interview

# Run migrations
alembic upgrade head

# Verify tables were created
psql -U your_username -d futuregen_db -c "\dt"
```

Listing 6: Run Database Migrations

5.2 Frontend Setup

```
# Install all npm packages
npm install

# Verify installation
npm list react
npm list @testing-library/react
```

Listing 7: Install Node.js Dependencies

6 Database Schema

6.1 Users Table

Column	Type	Constraints	Description
id	Integer	Primary Key, Auto Increment	Unique user identifier
user_name	String(100)	Not Null	Display name for the user
email	String(255)	Unique, Not Null	User's email address (login)
password	String(255)	Not Null	Hashed password using bcrypt
is_admin	Boolean	Default: False	Administrative privileges flag
created_at	DateTime	Default: Now	Account creation timestamp
updated_at	DateTime	Default: Now, Auto Update	Last modification timestamp
is_active	Boolean	Default: True	Account status flag

6.2 Registrations Table

Column	Type	Constraints	Description
id	Integer	Primary Key, Auto Increment	Unique registration identifier
name	String(200)	Not Null	Applicant's full name
email	String(255)	Not Null	Applicant's email address
registration_id	String(50)	Unique, Not Null	System-generated ID
status	String(50)	Default: 'pending'	Processing status
submitted_at	DateTime	Default: Now	Submission timestamp
resume_data	JSONB	Nullable	Parsed resume information
interview_data	JSONB	Nullable	Interview responses and scores
resume_comparison	JSONB	Nullable	Resume analysis results
created_at	DateTime	Default: Now	Record creation timestamp
updated_at	DateTime	Auto Update	Last modification timestamp

7 User Management

7.1 Creating Admin User

7.1.1 Using Seed Script

```
# Run the seed script
python seed_users.py

# This creates:
# - Admin user: admin@futuregen.com / admin123
# - Regular user: user@futuregen.com / user123
```

Listing 8: Automated User Creation

7.1.2 Manual API Creation

```
curl -X POST "http://localhost:8000/api/users/register" \
     -H "Content-Type: application/json" \
     -d '{
       "user_name": "System Administrator",
       "email": "admin@futuregen.com",
       "password": "SecureAdminPassword123!",
       "is_admin": true
     }'
```

Listing 9: Create Admin via API

8 Running the Application

8.1 Development Environment

8.1.1 Start Backend Server

```
# Ensure virtual environment is activated
source venv/bin/activate  # or venv\Scripts\activate on Windows

# Start FastAPI server with hot reload
uvicorn backend.main:app --reload --host 0.0.0.0 --port 8000

# Server will be available at:
# API: http://localhost:8000
# Swagger UI: http://localhost:8000/docs
# ReDoc: http://localhost:8000/redoc
```

Listing 10: Backend Development Server

8.1.2 Start Frontend Server

```
# In a new terminal, start React development server
npm start

# Frontend will be available at:
# http://localhost:3000
```

Listing 11: Frontend Development Server

9 Troubleshooting

9.1 Common Database Issues

Database Connection Errors

Issues related to database connectivity and configuration.

PostgreSQL Service Not Running

```
# Check PostgreSQL status
systemctl status postgresql  # Linux
brew services list | grep postgresql  # macOS
sc query postgresql-x64-13  # Windows

# Start PostgreSQL service
systemctl start postgresql  # Linux
brew services start postgresql  # macOS
net start postgresql-x64-13  # Windows
```

Database Connection Refused

- Verify PostgreSQL is running on the correct port (default 5432)
- Check `.env` file for correct database credentials
- Ensure database `futuregen_db` exists
- Verify user permissions on the database

Migration Errors

```
# Reset migrations if needed
alembic stamp head
alembic revision --autogenerate -m "Reset database"
alembic upgrade head

# Check current migration status
alembic current
alembic history
```

9.2 Python Environment Issues

Package Installation Errors

```
# Upgrade pip first
python -m pip install --upgrade pip

# Clear pip cache if needed
pip cache purge

# Install with verbose output for debugging
pip install -r requirements.txt -v
```

Virtual Environment Issues

```
# Recreate virtual environment
deactivate  # if currently activated
rm -rf venv  # remove old environment
python -m venv venv  # create new environment
source venv/bin/activate  # activate
pip install -r requirements.txt  # reinstall packages
```

9.3 Frontend Connection Issues

CORS Errors

- Verify backend CORS configuration in `main.py`
- Check that frontend URL is in `CORS_ORIGINS` list
- Ensure both servers are running on expected ports

Proxy Configuration Add to `package.json` if needed:

```
{
  "name": "futuregen-frontend",
  "proxy": "http://localhost:8000",
  ...
}
```

10 Maintenance Scripts

10.1 Database Management

Script	Description
`python create_new_tables.py`	Creates or updates database tables as per schema changes
`python seed_users.py`	Seeds initial admin and user accounts for development

10.2 Development Utilities

Command	Description
`uvicorn backend.main:app --reload`	Starts backend server in development mode with hot reload
`npm install`	Installs frontend dependencies from package.json
`npm start`	Starts frontend development server with hot reload

End of Documentation