

Jobs.AI - Installation and Execution Guide

Prerequisites

- Node.js (v16 or higher)
- npm (v8 or higher)
- MongoDB (v4.4 or higher)

Project Structure

```
jobs.ai/  
% % %  backend/           # Backend server  
% % %  src/               # Frontend React application  
% % %  public/            # Static files  
% % %  package.json       # Frontend dependencies
```

Installation Steps

1. Frontend Setup

1. Open a terminal and navigate to the project root directory:

```
cd /path/to/jobs.ai
```

2. Install dependencies:

```
npm install
```

```
,
```

2. Backend Setup

1. Navigate to the backend directory:

```
cd backend
```

2. Install dependencies:

```
npm install
```

Running the Application

1. Start the Backend Server

1. In the backend directory, start the development server:

```
npm run dev
```

The backend server will start on <http://localhost:5000>

2. Start the Frontend Application

1. Open a new terminal window

2. Navigate to the project root directory:

```
cd /path/to/jobs.ai
```

3. Start the React development server:

```
npm start
```

The frontend application will start on <http://localhost:3000>

Accessing the Application

1. Open your web browser and navigate to `http://localhost:3000`
2. You should see the Jobs.AI application interface

Available Scripts

Frontend Scripts

- ``npm start``: Runs the app in development mode
- ``npm run build``: Builds the app for production
- ``npm test``: Launches the test runner
- ``npm run eject``: Ejects from Create React App

Backend Scripts

- ``npm start``: Runs the server in production mode
- ``npm run dev``: Runs the server in development mode with hot reloading

Troubleshooting

Common Issues

1. ****Port Already in Use****
 - If port 3000 or 5000 is already in use, you can change the port in the respective ``.env`` files
2. ****MongoDB Connection Issues****
 - Ensure MongoDB is running
 - Verify the connection string in the backend ``.env`` file
3. ****Missing Dependencies****
 - If you encounter missing module errors, try:

```
```bash
npm install
```
```
 - For backend dependencies:

```
```bash
cd backend
```

npm install  
...

4. **Environment Variables**

- Ensure all required environment variables are set in both `.env`` files
- Double-check the values for accuracy

**Development Notes**

- The frontend uses React with Material-UI for the interface
- The backend is built with Express.js
- MongoDB is used as the database
- Authentication is handled using JWT and Google OAuth
- PDF processing is handled using various PDF libraries
- OpenAI integration is used for AI-powered features

**Cloud Services Architecture**

**Overview**

The application utilizes multiple cloud services to provide a robust and scalable infrastructure:

Frontend	Backend	Database
(Firebase)	(App Engine)	(MongoDB)
Authentication	Storage	Analytics
(Firebase Auth)	(Cloud Storage)	(Firebase)

**Service Relationships**

1. **\*\*Frontend (Firebase Hosting)\*\***
  - Hosts the React application
  - Serves static files
  - Integrates with Firebase Authentication
  - Connects to backend API
2. **\*\*Backend (Google App Engine)\*\***
  - Handles API requests
  - Processes PDF files
  - Manages business logic
  - Connects to MongoDB
  - Integrates with OpenAI API
3. **\*\*Database (MongoDB Atlas)\*\***
  - Stores user data
  - Manages document metadata
  - Handles analytics data
  - Provides data persistence
4. **\*\*Authentication (Firebase Auth)\*\***
  - Manages user authentication
  - Handles Google OAuth
  - Provides JWT tokens
  - Integrates with frontend and backend
5. **\*\*Storage (Google Cloud Storage)\*\***
  - Stores PDF files
  - Manages document uploads
  - Handles file processing
  - Provides secure file access
6. **\*\*Analytics (Firebase Analytics)\*\***
  - Tracks user activity
  - Monitors application performance
  - Provides usage insights
  - Helps with debugging

## **Service Configuration**

### 1. **\*\*Firebase Configuration\*\***

```
```javascript
const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
  projectId: process.env.REACT_APP_FIREBASE_PROJECT_ID,
  storageBucket: process.env.REACT_APP_FIREBASE_STORAGE_BUCKET,
  messagingSenderId:
process.env.REACT_APP_FIREBASE_MESSAGING_SENDER_ID,

  appId: process.env.REACT_APP_FIREBASE_APP_ID
};
```
```

### 2. **\*\*MongoDB Connection\*\***

```
```javascript
const mongoURI = process.env.MONGODB_URI;
mongoose.connect(mongoURI, {
  useNewUrlParser: true,
  useUnifiedTopology: true
});
```
```

### 3. **\*\*Google Cloud Storage\*\***

```
```javascript
const storage = new Storage({
  projectId: process.env.GOOGLE_CLOUD_PROJECT_ID,
  keyFilename: process.env.GOOGLE_APPLICATION_CREDENTIALS
});
```
```

## **Environment Variables for Cloud Services**

Add these to your ``.env`` files:

# **Firebase**

`REACT_APP_FIREBASE_API_KEY=your_firebase_api_key`

REACT\_APP\_FIREBASE\_AUTH\_DOMAIN=your\_firebase\_auth\_domain

REACT\_APP\_FIREBASE\_PROJECT\_ID=your\_firebase\_project\_id

REACT\_APP\_FIREBASE\_STORAGE\_BUCKET=your\_firebase\_storage\_bucket

REACT\_APP\_FIREBASE\_MESSAGING\_SENDER\_ID=your\_firebase\_messaging\_sender\_id

REACT\_APP\_FIREBASE\_APP\_ID=your\_firebase\_app\_id

## Google Cloud

GOOGLE\_CLOUD\_PROJECT\_ID=your\_google\_cloud\_project\_id

GOOGLE\_APPLICATION\_CREDENTIALS=path\_to\_your\_credentials.json

## MongoDB

MONGODB\_URI=your\_mongodb\_connection\_string

### Deployment Considerations

#### 1. **Frontend Deployment**

- Deploy to Firebase Hosting
- Configure build settings
- Set up environment variables
- Enable HTTPS

#### 2. **Backend Deployment**

- Deploy to Google App Engine
- Configure scaling settings
- Set up environment variables
- Enable monitoring

#### 3. **Database Setup**

- Configure MongoDB Atlas
- Set up network access
- Configure backup settings
- Monitor performance

#### 4. **Storage Configuration**

- Set up Google Cloud Storage
- Configure CORS settings
- Set up access control
- Enable versioning

#### 5. **\*\*Authentication Setup\*\***

- Configure Firebase Auth
- Set up OAuth providers
- Configure security rules
- Enable email verification

## **Monitoring and Maintenance**

#### 1. **\*\*Performance Monitoring\*\***

- Use Firebase Analytics
- Monitor API response times
- Track resource usage
- Set up alerts

#### 2. **\*\*Security Monitoring\*\***

- Monitor authentication attempts
- Track API usage
- Monitor storage access
- Set up security alerts

#### 3. **\*\*Cost Management\*\***

- Monitor service usage
- Set up budget alerts
- Optimize resource usage
- Review billing reports

## **Security Considerations**

1. Never commit `.env` files to version control
2. Keep your API keys and secrets secure
3. Use environment variables for sensitive information
4. Regularly update dependencies to patch security vulnerabilities

## **Support**



If you encounter any issues or need assistance:

1. Check the troubleshooting section
2. Review the error messages in the console
3. Check the browser's developer tools for frontend issues
4. Review the server logs for backend issues

---