VidyaMitra: The Intelligent Career Agent

AI-Powered Resume Evaluator, Trainer & Career Planner

Project Description:

The VidyaMitra platform is a modern web-based application that enhances digital learning by integrating artificial intelligence to support students and educators with personalized educational assistance. It provides easy access to relevant content, smart recommendations, and a user-friendly interface that improves engagement and learning effectiveness.

The system uses a full-stack architecture consisting of a React.js frontend for dynamic and responsive user interaction, and a Python backend developed with FastAPI for handling APIs, authentication, data processing, and AI functionalities. A dedicated virtual environment is used to isolate backend dependencies, and the frontend and backend are run in separate terminals to ensure smooth and independent execution.

The development setup supports both Windows and macOS through the use of VS Code terminals for running services and managing the project efficiently. With a scalable structure and AI-driven features, VidyaMitra can continue to grow with advanced analytics, improved content access, and cloud deployment options, making it a strong and adaptable solution for digital education.

Scenario 1: Personalized Resume Evaluation and Skill Mapping

A final-year engineering student uploads her resume to VidyaMitra for AI evaluation. The system identifies gaps like data visualization and cloud fundamentals and suggests courses to enhance her employability.

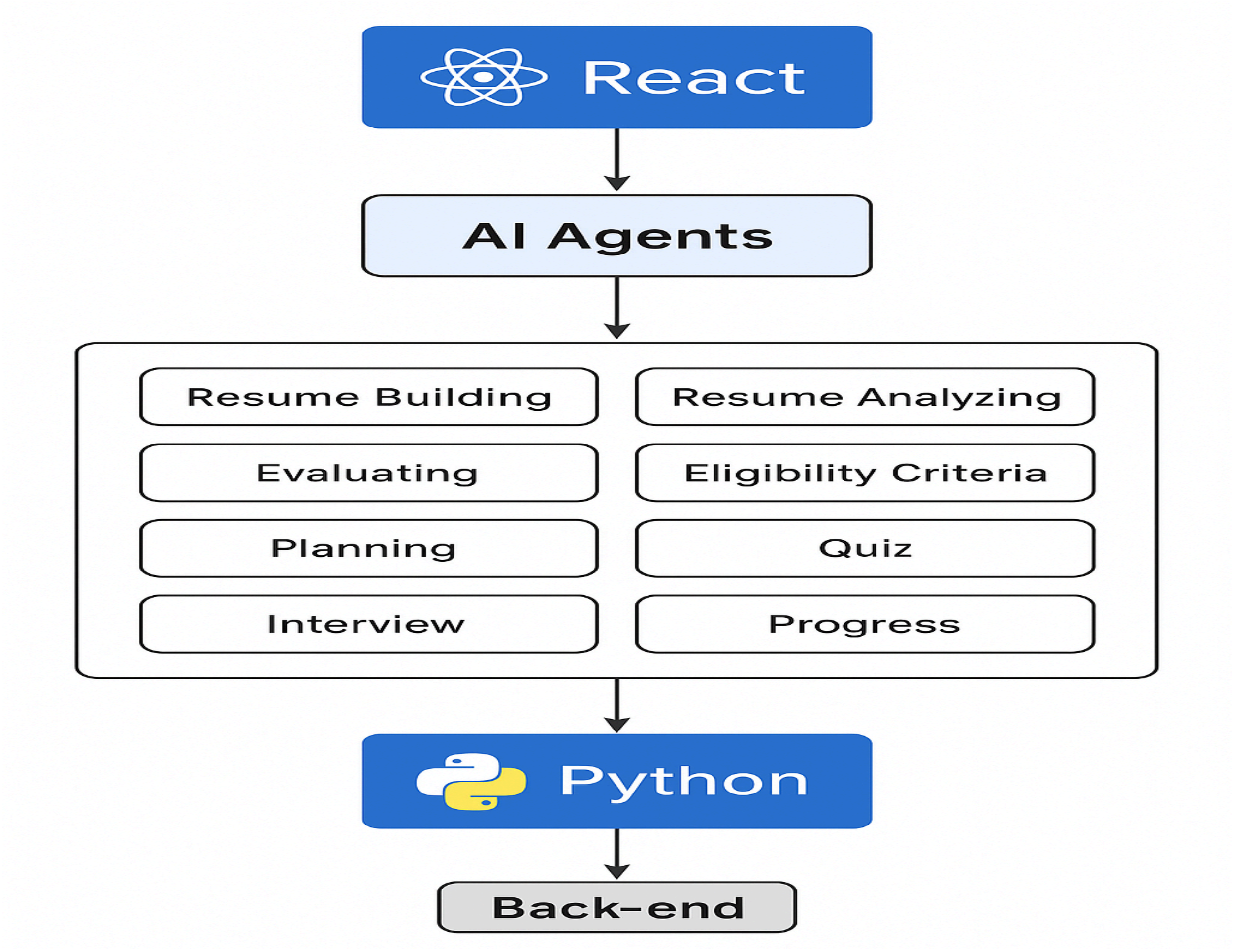
Scenario 2: AI-Driven Mock Interview and Feedback Generation

An MBA graduate uses VidyaMitra's AI interview simulator for realistic mock interviews, receiving feedback and personalized guidance to improve communication and interview readiness. The system evaluates tone, confidence, and accuracy for focused improvement.

Scenario 3: Career Path Recommendation and Upskilling Planner

A working professional exploring a shift to data science uses VidyaMitra to analyze his resume and skills. The system identifies transferable strengths, recommends learning paths and certifications, and creates a personalized roadmap for a smooth career transition.

Architecture:



Pre-requisites:

1. **Python Environment Setup:** Install Python 3.10+ and create a virtual environment.
   https://www.python.org/downloads/

2. **VS Code Setup:** Download it from the official site:
   https://code.visualstudio.com/download

3. **Node.js Installation:** Install Node.js 18+ (includes npm 9+)
   https://nodejs.org/en/download.

4. **Backend Dependencies:** Install required Python packages using pip install -r requirements.txt.
   https://pip.pypa.io/en/stable/cli/pip_install/

5. **Frontend Dependencies:** Install JavaScript packages using npm install.
   https://docs.npmjs.com/cli/v9/commands/npm-install

6. **Database Setup:** VidyāMitra supports **Supabase** for managing and storing of data.
   https://www.sqlite.org/download.html

7. **Backend Configuration:** Add .env file with API keys and database URL.
   https://pypi.org/project/python-dotenv/

8. **Frontend Configuration:** In the frontend root folder, create a `.env` file
   https://fastapi.tiangolo.com/

9. **Backend Launch:** Start FastAPI server with python -m uvicorn main:app --reload.
   https://www.uvicorn.org/

10. **Frontend Launch:** Run development server with npm run dev.
    https://docs.npmjs.com/cli/v9/commands/npm-run-script

11. **Verification:** Once both servers are running successfully:

    API Documentation: http://localhost:8000/docs
    Frontend Interface: http://localhost:5173

Project WorkFlow:

Milestone 1: Environment Setup and Project Initialization

Activity 1.1: Create and activate a virtual environment.
Activity 1.2: Set up project folder structure for backend and frontend.
Activity 1.3: Configure backend and frontend .env files with credentials.
Activity 1.4: Install require dependencies.
Activity 1.5: Test setup by running backend and frontend servers.

Milestone 2: Backend API Development using FastAPI

Activity 2.1: Create secure user authentication endpoints for registration and login.
Activity 2.2: Create modular routers for all backend functions.

Milestone 3: AI Integration with OpenAI, LangChain, and External APIs

Activity 3.1: Connect GPT-4 API using OPENAI_API_KEY for career support.
Activity 3.2: Use Google and YouTube APIs with keys to fetch personalized learning resources.
Activity 3.3: Use Supabase for secure cloud storage and real-time sync.
Activity 3.4: Use Pexels API with PEXELS_API_KEY to add visual learning resources.
Activity 3.5: Fetch live market and financial updates via News and Exchange APIs.

Milestone 4: React.js Frontend Development

Activity 4.1: Create a responsive dashboard for all career modules.
Activity 4.2: Create interactive pages for resume, skills, training, and interviews.

Milestone 5: Testing and Deployment

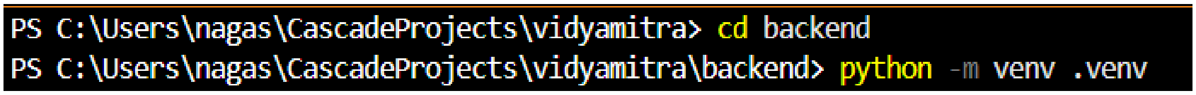
Activity 5.1: Backend Testing and API Validation
Activity 5.2: Frontend Functionality Testing
Activity 5.3: Backend–Frontend Integration Testing
Activity 5.4: Output Validation and User Experience Review

Milestone 1: Environment Setup and Project Initialization

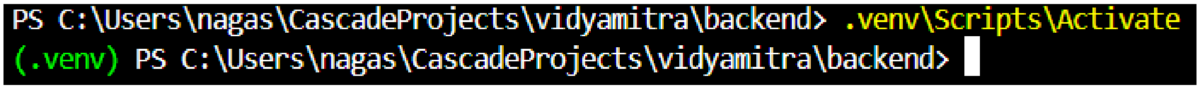
- **Activity 1.1: Create and activate a virtual environment.**

  Create and activate a Python virtual environment using python -m venv venv.

  To begin the backend setup, a terminal must be opened such as Command Prompt in Windows (Win + R → type cmd → Enter) or the integrated terminal in VS Code (Ctrl + ~). Once the terminal is open, you navigate into the backend folder of the project using the command "cd backend," ensuring all backend configurations are performed in the correct directory. Inside this folder, a dedicated virtual environment is created using "python -m venv .venv," which isolates all project-specific Python libraries from the global installation to prevent version conflicts and maintain consistency across different systems. After the virtual environment is created, it is activated with the command ".venv\Scripts\Activate" on Windows, after which the terminal will show a prefix like "(.venv)" to indicate that the environment is active. From this point onward, every library installation and backend operation will run safely inside this isolated workspace, ensuring stable and organized dependency management throughout the development process.



```
PS C:\Users\nagas\CascadeProjects\vidyamitra> cd backend
PS C:\Users\nagas\CascadeProjects\vidyamitra\backend> python -m venv .venv
```

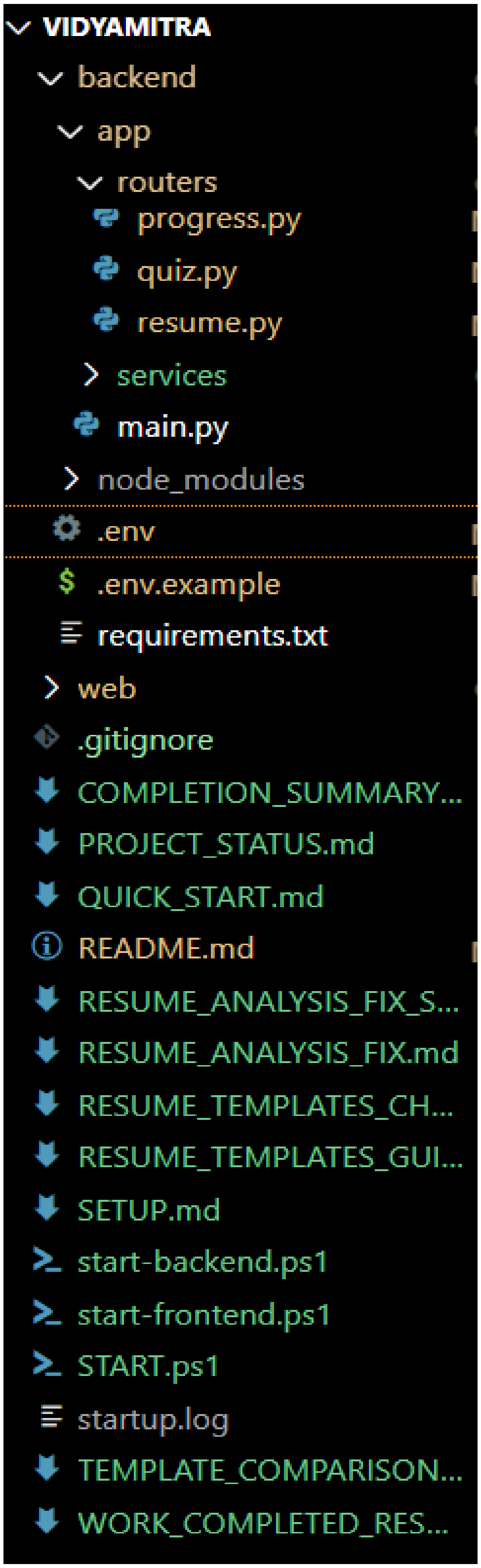


```
PS C:\Users\nagas\CascadeProjects\vidyamitra\backend> .venv\Scripts\Activate
(.venv) PS C:\Users\nagas\CascadeProjects\vidyamitra\backend>
```

- **Activity 1.2: Set up project folder structure for backend and frontend.**

  Set up project folder structure for backend (FastAPI) and frontend (React.js).

  A well-structured directory layout is established to organize the project efficiently. The backend folder is dedicated to FastAPI application files, including API routes, models, and configuration scripts, while the frontend folder(web here) contains the React.js code for the user interface. This separation promotes modularity, easier debugging, and parallel development between the backend and frontend teams.

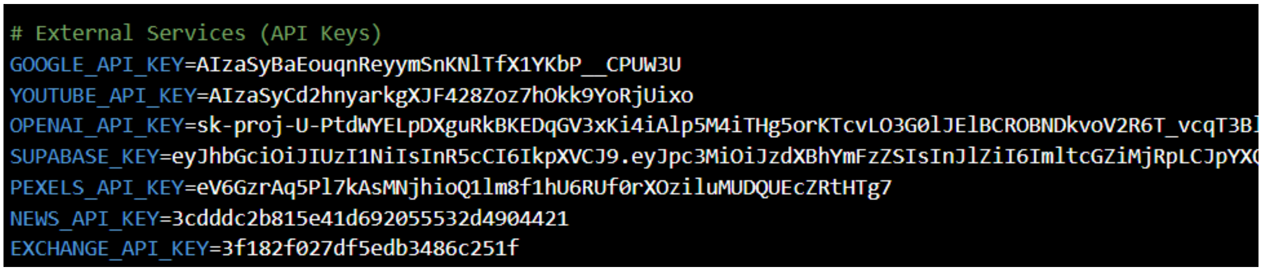


```
VIDYAMITRA
  backend
    app
      routers
        progress.py
        quiz.py
        resume.py
      services
      main.py
    node_modules
    .env
    .env.example
    requirements.txt
  web
  .gitignore
  COMPLETION_SUMMARY...
  PROJECT_STATUS.md
  QUICK_START.md
  README.md
  RESUME_ANALYSIS_FIX_S...
  RESUME_ANALYSIS_FIX.md
  RESUME_TEMPLATES_CH...
  RESUME_TEMPLATES_GUI...
  SETUP.md
  start-backend.ps1
  start-frontend.ps1
  START.ps1
  startup.log
  TEMPLATE_COMPARISON...
  WORK_COMPLETED_RES...
```

- **Activity 1.3: Configure backend and frontend .env files with credentials.**

  Configure .env files for both backend and frontend with required API keys and credentials.

  Environment variables are managed using .env files to securely store sensitive data like API keys, database connection URLs, and authentication secrets. These files help maintain security and flexibility by keeping configuration details outside the main source code. Separate .env files are used for the backend and frontend, ensuring that both environments can be configured independently as needed.

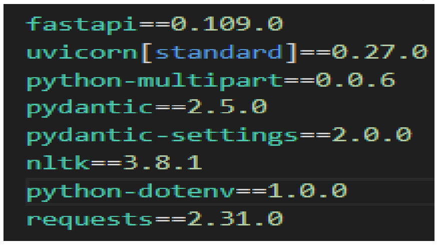


```
# External Services (API Keys)
GOOGLE_API_KEY=AIzaSyBaEouqnReyymSnKNlTfX1YKbP__CPUW3U
YOUTUBE_API_KEY=AIzaSyCd2hnyarkgXJF428Zoz7hOkk9YoRjUixo
OPENAI_API_KEY=sk-proj-U-PtdWYELpDXguRkBKEDqGV3xKi4iAlp5M4iTHg5orKTcvLO3G0lJElBCROBNDkvoV2R6T_vcqT3B
SUPABASE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImltcGZiMjRpLCJpYX
PEXELS_API_KEY=eV6GzrAq5Pl7kAsMNjhioQ1lm8f1hU6RUf0rXOziluMUDQUEcZRtHTg7
NEWS_API_KEY=3cdddc2b815e41d692055532d4904421
EXCHANGE_API_KEY=3f182f027df5edb3486c251f
```

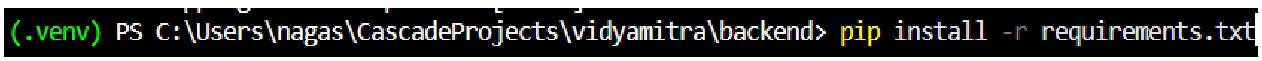
- **Activity 1.4: Install require dependencies**

  Install dependencies using pip install -r requirements.txt and npm install.

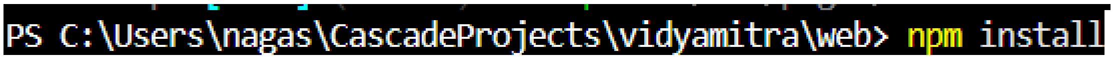
  In this step, all required dependencies and libraries are installed for both the backend and frontend environments. The backend dependencies, listed in the requirements.txt file, include FastAPI, and other essential Python packages are installed with the command "pip install -r requirements.txt". The frontend dependencies are installed using "npm install" to set up React.js, Tailwind CSS, and related UI libraries. This ensures that the complete development environment is ready with all necessary tools.



```
fastapi==0.109.0
uvicorn[standard]==0.27.0
python-multipart==0.0.6
pydantic==2.5.0
pydantic-settings==2.0.0
nltk==3.8.1
python-dotenv==1.0.0
requests==2.31.0
```



```
(.venv) PS C:\Users\nagas\CascadeProjects\vidyamitra\backend> pip install -r requirements.txt
```



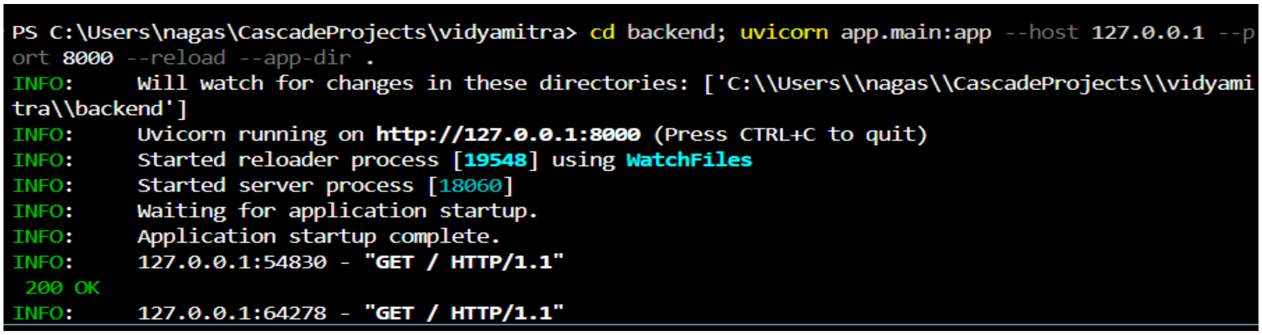
```
PS C:\Users\nagas\CascadeProjects\vidyamitra\web> npm install
```

- **Activity 1.5: Test setup by running backend and frontend servers.**

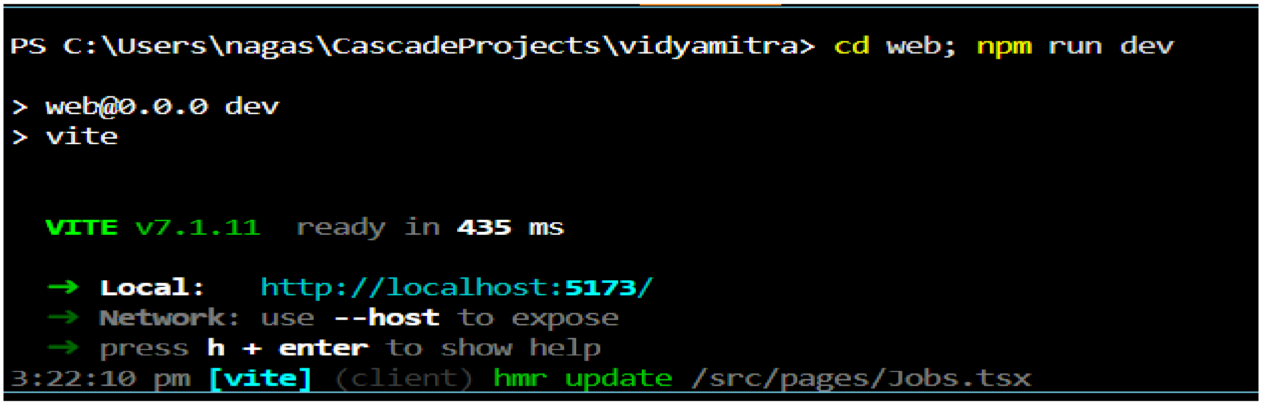
Verify environment setup by launching both backend and frontend development servers.

To confirm successful setup, both servers are started locally : the backend with “uvicorn app.main:app --reload” and the frontend with “npm run dev.” The backend API can be verified through its interactive Swagger documentation at http://127.0.0.1:8000, while the frontend interface is accessible at http://localhost:5173/ This final activity ensures that both components are properly connected and operational, marking the completion of the environment setup phase.

The backend and frontend should be run in separate terminals to ensure they operate independently, allowing each service to function smoothly without interrupting or interfering with the other.



```
PS C:\Users\nagas\CascadeProjects\vidyamitra> cd backend; uvicorn app.main:app --host 127.0.0.1 --p
ort 8000 --reload --app-dir .
INFO:     Will watch for changes in these directories: ['C:\\Users\\nagas\\CascadeProjects\\vidyami
tra\\backend']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [19548] using WatchFiles
INFO:     Started server process [18060]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:54830 - "GET / HTTP/1.1"
 200 OK
INFO:     127.0.0.1:64278 - "GET / HTTP/1.1"
```



```
PS C:\Users\nagas\CascadeProjects\vidyamitra> cd web; npm run dev

> web@0.0.0 dev
> vite


  VITE v7.1.11  ready in 435 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
3:22:10 pm [vite] (client) hmr update /src/pages/Jobs.tsx
```

Conclusion:

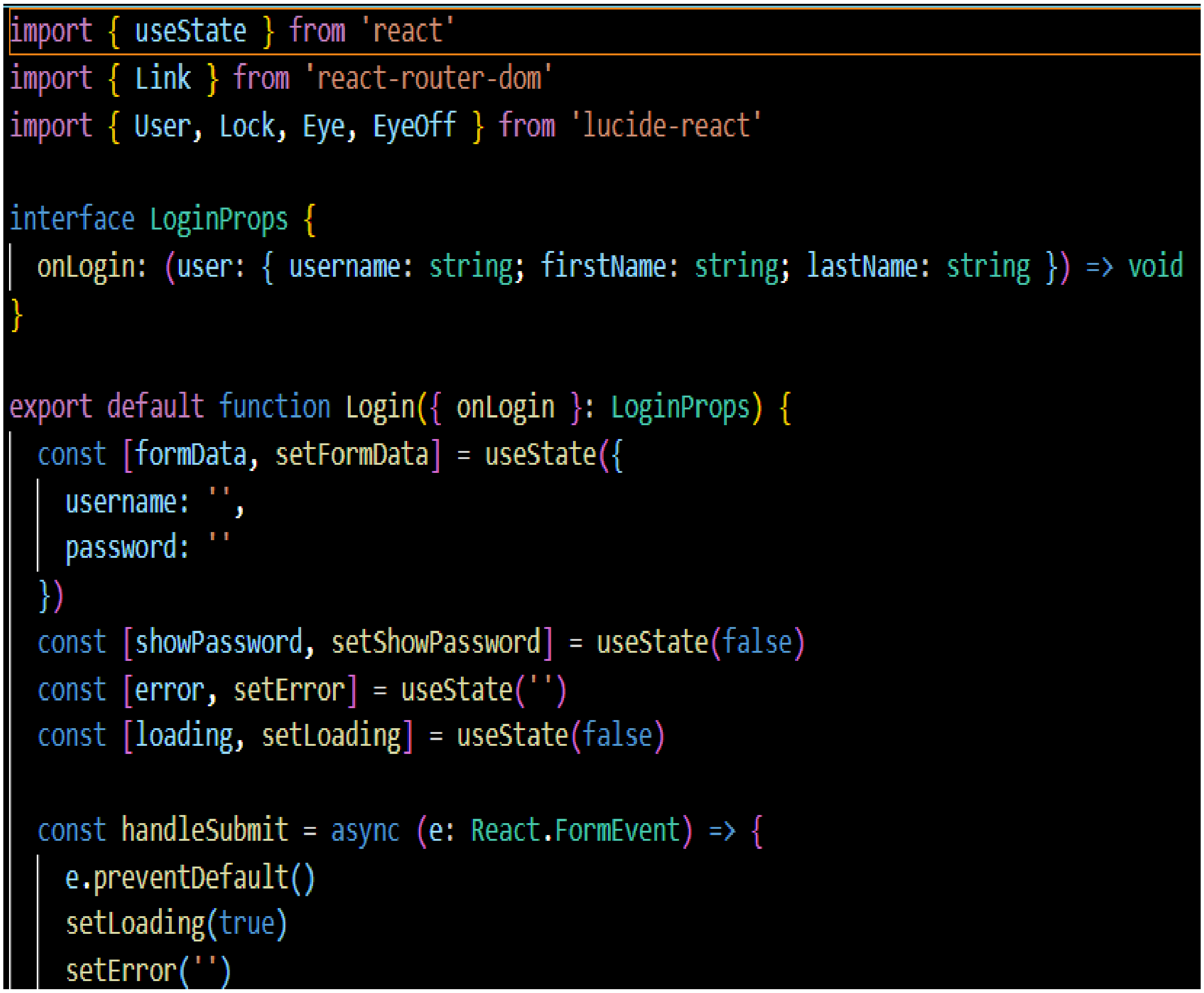
Milestone 1 focuses on setting up the development environment for the VidyaMitra project. It includes creating a Python virtual environment, organizing backend (FastAPI) and frontend (React.js) folders, configuring secure .env files, installing all required dependencies, and verifying the setup by running both servers. This ensures a fully isolated, modular, and operational environment ready for development.

Milestone 2: Backend API Development using FastAPI

- **Activity 2.1: Create secure user authentication endpoints for registration and login.**

  Develop user authentication endpoints for secure registration and login using and password page

  This React code defines a Login component that manages user authentication input and state. It uses the `useState` hook to store form data for the username and password, handle password visibility, and manage loading and error states during submission. The `handleSubmit` function prevents the default form behavior and triggers an asynchronous login process while updating UI states. The component also imports icons for better UI feedback and integrates with React Router for navigation. Overall, it provides an interactive, state-managed login form for secure user authentication in the VidyāMitra frontend.

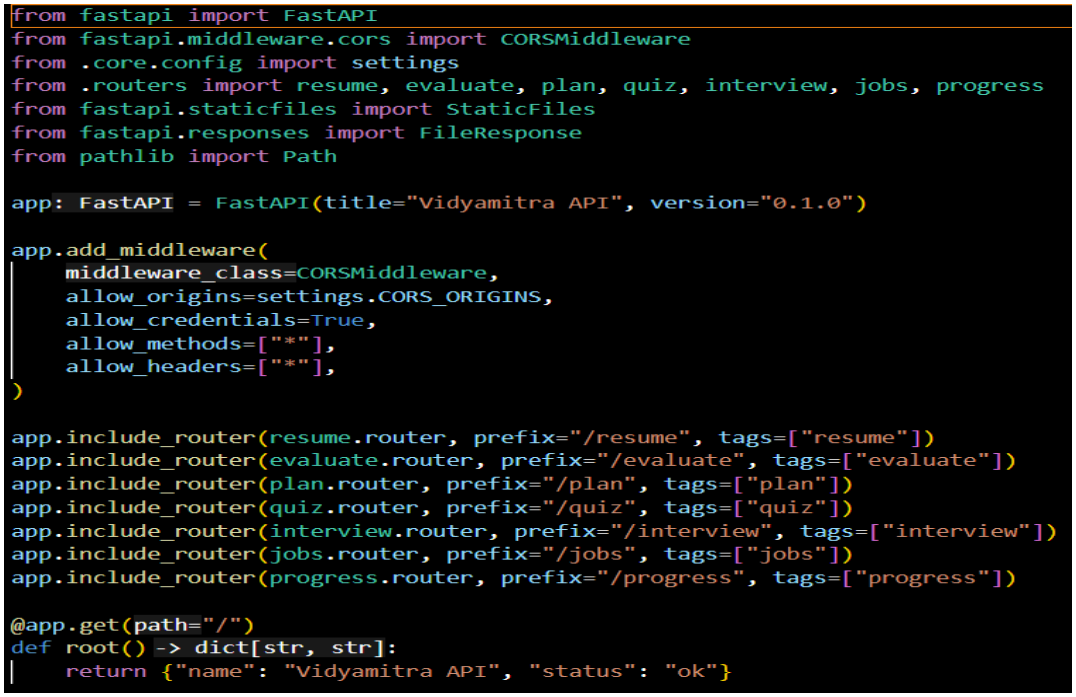


```
import { useState } from 'react'
import { Link } from 'react-router-dom'
import { User, Lock, Eye, EyeOff } from 'lucide-react'

interface LoginProps {
  onLogin: (user: { username: string; firstName: string; lastName: string }) => void
}

export default function Login({ onLogin }: LoginProps) {
  const [formData, setFormData] = useState({
    username: '',
    password: ''
  })
  const [showPassword, setShowPassword] = useState(false)
  const [error, setError] = useState('')
  const [loading, setLoading] = useState(false)

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault()
    setLoading(true)
    setError('')
```

- **Activity 2.2: Create modular routers for all backend functions.**

Create modular routers for each backend function Resume Parsing, Skill Evaluation, Training Plan Generation, Quiz & Interview Simulation, and Progress Tracking.

This FastAPI script serves as the backend entry point for the VidyāMitra API, defining the core server configuration and route integration. It initializes a `FastAPI` instance with metadata like the project title and version, then adds CORS middleware to allow secure communication between the backend and React frontend. The script organizes the application into modular routers for key features such as resume parsing, skill evaluation, training plan generation, quizzes, mock interviews, job recommendations, and progress tracking. Each router is assigned a unique URL prefix and tag for better API documentation and modularity. A simple root endpoint (`"/"`) is also defined to confirm the server's running status. Overall, this file ensures a scalable, organized, and well-structured backend architecture for VidyāMitra's AI-powered career ecosystem.



```python
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from .core.config import settings
from .routers import resume, evaluate, plan, quiz, interview, jobs, progress
from fastapi.staticfiles import StaticFiles
from fastapi.responses import FileResponse
from pathlib import Path

app: FastAPI = FastAPI(title="Vidyamitra API", version="0.1.0")

app.add_middleware(
    middleware_class=CORSMiddleware,
    allow_origins=settings.CORS_ORIGINS,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.include_router(resume.router, prefix="/resume", tags=["resume"])
app.include_router(evaluate.router, prefix="/evaluate", tags=["evaluate"])
app.include_router(plan.router, prefix="/plan", tags=["plan"])
app.include_router(quiz.router, prefix="/quiz", tags=["quiz"])
app.include_router(interview.router, prefix="/interview", tags=["interview"])
app.include_router(jobs.router, prefix="/jobs", tags=["jobs"])
app.include_router(progress.router, prefix="/progress", tags=["progress"])

@app.get(path="/")
def root() -> dict[str, str]:
    return {"name": "Vidyamitra API", "status": "ok"}
```

Conclusion:

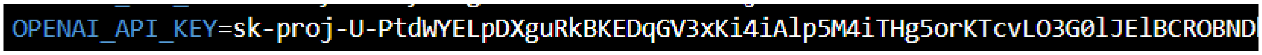
Milestone 2 involves developing the backend APIs for VidyaMitra using FastAPI. It includes creating secure user authentication endpoints for registration and login, as well as organizing core functionalities like resume parsing, skill evaluation, training plan generation, quizzes, mock interviews, and progress tracking into modular routers. This structure ensures a scalable, secure, and well-organized backend to support the platform's AI-powered career services.

Milestone 3: AI Integration with OpenAI, LangChain, and External APIs

- **Activity 3.1: Connect GPT-4 API using OPENAI_API_KEY for career support.**

  Integrate OpenAI GPT-4 API using OPENAI_API_KEY for intelligent resume analysis, interview simulation, and personalized career feedback generation.

  This activity focuses on integrating the OpenAI GPT-4 API into the VidyāMitra backend to enable intelligent career insights and personalized assistance. Using the OPENAI_API_KEY, the system connects securely to OpenAI's API to analyze resumes, simulate interview questions, and generate tailored feedback for users. When a user uploads their resume or starts a mock interview, the backend sends the data to the GPT-4 model for processing, and the AI returns structured evaluations and recommendations. This integration enhances the platform's ability to provide dynamic, context-aware guidance for students and job seekers.
  https://platform.openai.com/api-keys



```
OPENAI_API_KEY=sk-proj-U-PtdWYELpDXguRkBKEDqGV3xKi4iAlp5M4iTHg5orKTcvLO3G0lJElBCROBND
```

1. Go to the link above.

2. Log in with your OpenAI account (or sign up if you don't have one).

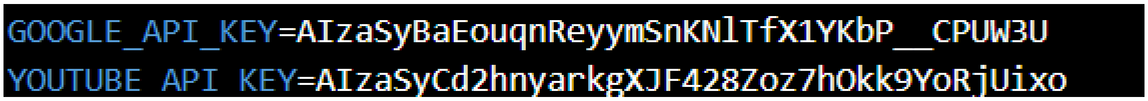
3. Click **"Create new secret key."**

4. Copy the generated key

- **Activity3.2: Use Google and YouTube APIs with keys to fetch personalized learning resources.**

  Connect Google and YouTube APIs (GOOGLE_API_KEY, YOUTUBE_API_KEY) to fetch top-rated learning materials, tutorials, and domain-related videos that align with each user's personalized training plan.

  This activity integrates the Google and YouTube APIs into the VidyāMitra platform to automatically fetch high-quality, domain-specific learning resources. Using the GOOGLE_API_KEY and YOUTUBE_API_KEY, the backend connects to YouTube's Data API to retrieve top-rated tutorials, lectures, and skill-building videos relevant to each user's personalized training plan. When the AI identifies a user's skill gaps, it queries YouTube for matching educational content and displays it directly within the dashboard. This enhances user engagement and ensures tailored learning support based on their career goals.
  https://console.cloud.google.com/apis/credentials



```
GOOGLE_API_KEY=AIzaSyBaEouqnReyymSnKNlTfX1YKbP__CPUW3U
YOUTUBE API KEY=AIzaSyCd2hnyarkgXJF428Zoz7hOkk9YoRjUixo
```

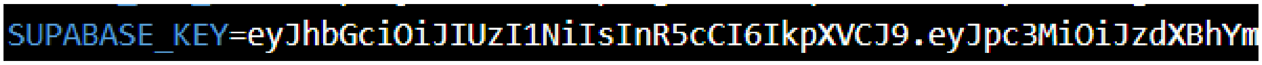
1. Sign in to your Google account.
2. Create a new project in Google Cloud Console.
3. Enable **YouTube Data API v3** under "APIs & Services → Library."
4. Go to **APIs & Services** → **Credentials** → **Create Credentials** → **API Key.**
5. Copy the generated key and add it to your `.env`

- **Activity 3.3: Use Supabase for secure cloud storage and real-time sync.**

  Use Supabase (SUPABASE_KEY) as a secure cloud-based data layer for storing user profiles, skill analytics, and progress tracking, enabling real-time data synchronization across all agents.

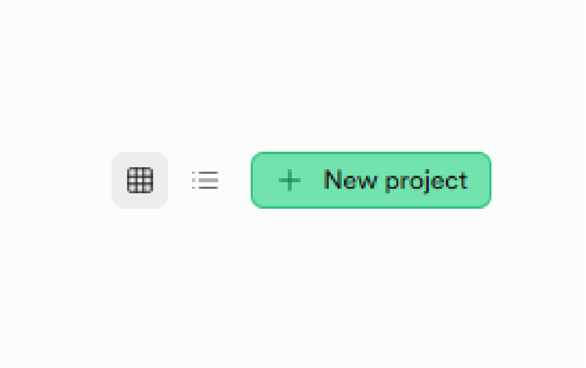
  Supabase is used in this project to securely store and manage user data, authentication details, and learning information in a scalable database. It allows the backend to perform real-time data operations through secure API communication, ensuring smooth and efficient functioning of the application.

  This activity involves integrating Supabase into the VidyāMitra platform as a secure, scalable, and real-time data management layer. Using the SUPABASE_KEY, the backend connects to Supabase to store and retrieve essential data such as user profiles, skill analytics, resume scores, and training progress. Supabase provides a PostgreSQL-backed database with built-in authentication, APIs, and real-time event listeners, ensuring seamless synchronization between all AI agents including ResumeAgent, EvaluatorAgent, and ProgressAgent. This integration allows data updates (like skill improvements or quiz results) to reflect instantly across the system, enhancing interactivity and consistency.
  https://app.supabase.com/project

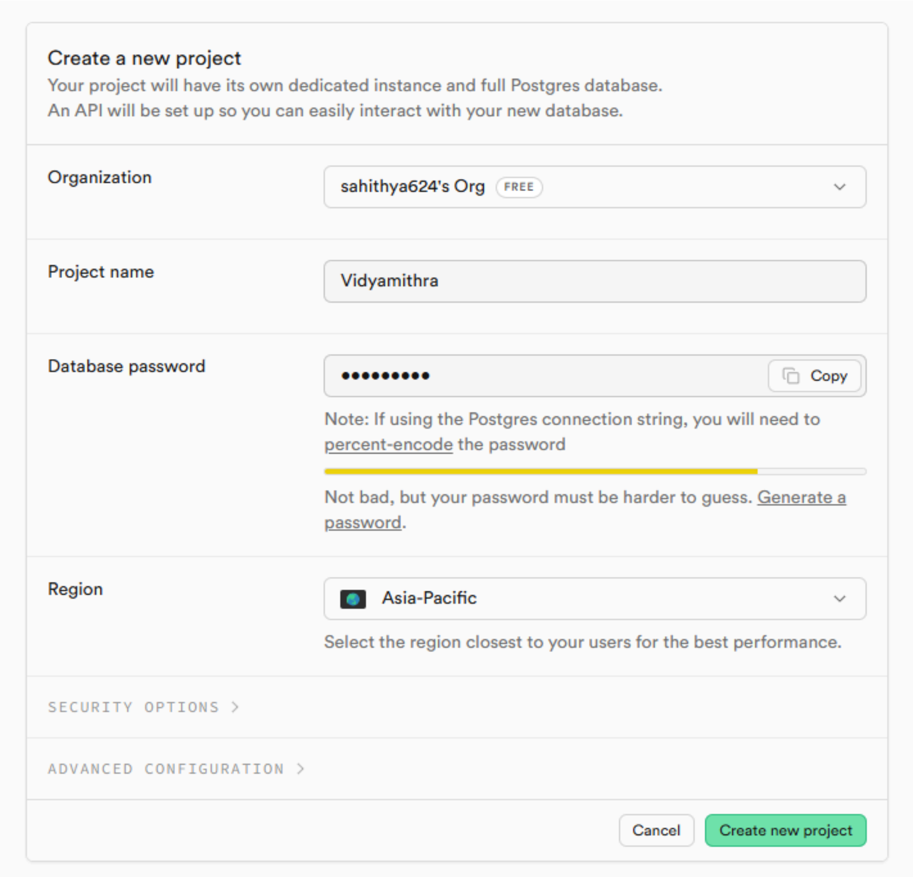


```
SUPABASE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYm
```

1. Sign in or create an account on Supabase.
2. Click "New Project" and give it a name (e.g., *VidyāMitra*).



3. Choose a database password and region, then click Create Project.



Create a new project

Your project will have its own dedicated instance and full Postgres database.
An API will be set up so you can easily interact with your new database.

Organization: sahithya624's Org FREE

Project name: Vidyamithra

Database password: ●●●●●●●●●● Copy

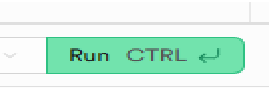
Note: If using the Postgres connection string, you will need to percent-encode the password

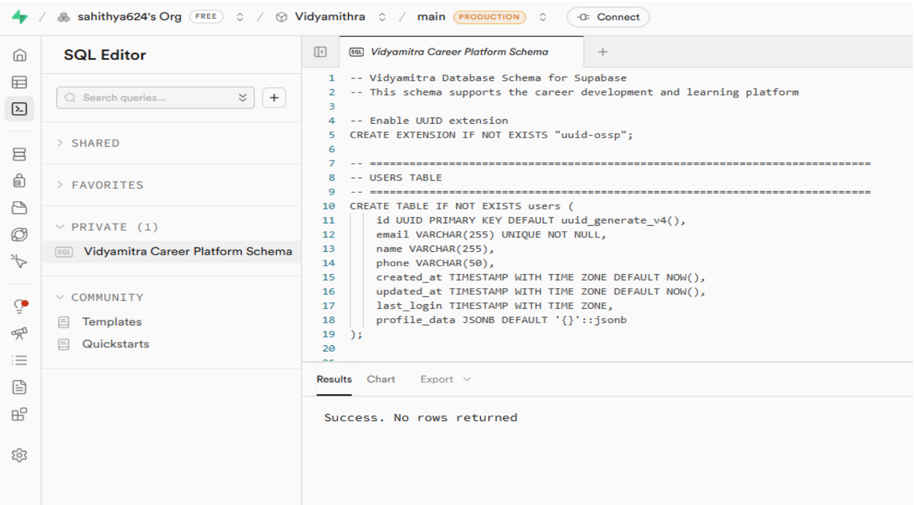
Not bad, but your password must be harder to guess. Generate a password.

Region: Asia-Pacific

Select the region closest to your users for the best performance.

4. Now, Go to SQL Editor ⟶ Add schema.sql ⟶ Click on Run Ctrl





```sql
-- Vidyamitra Database Schema for Supabase
-- This schema supports the career development and learning platform

-- Enable UUID extension
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

-- ============================================================================
-- USERS TABLE
-- ============================================================================
CREATE TABLE IF NOT EXISTS users (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    email VARCHAR(255) UNIQUE NOT NULL,
    name VARCHAR(255),
    phone VARCHAR(50),
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    last_login TIMESTAMP WITH TIME ZONE,
    profile_data JSONB DEFAULT '{}'::jsonb
);
```

```
Success. No rows returned
```

5. Once table is created, You will see a " Success. No rows returned"

6. Now, Go to Table Editor → You will see the tables which are created.

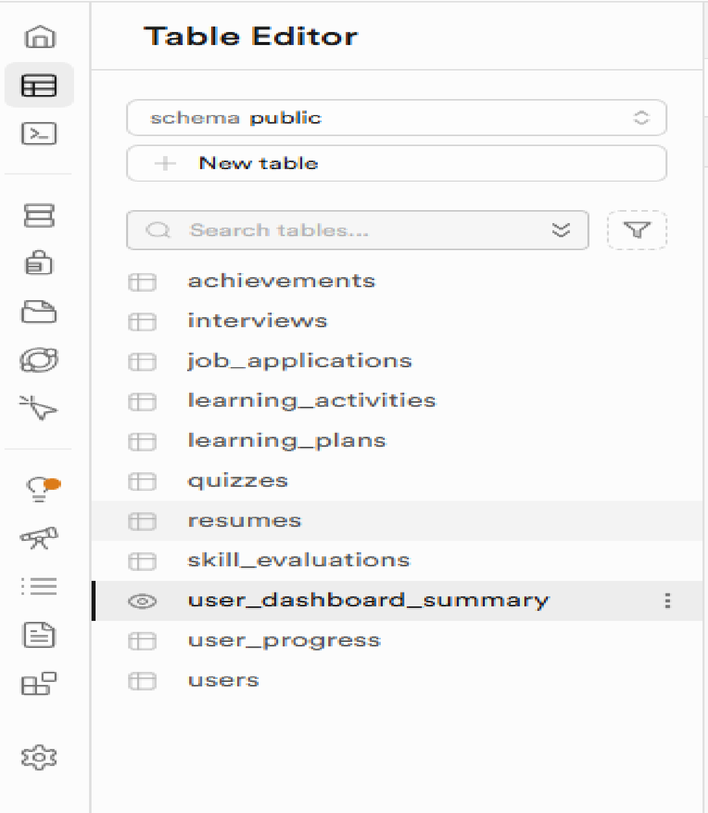


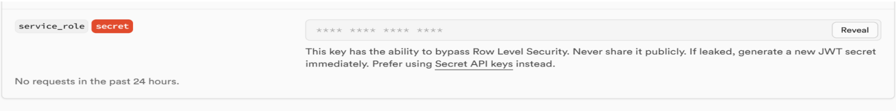
Table Editor

schema public

achievements
interviews
job_applications
learning_activities
learning_plans
quizzes
resumes
skill_evaluations
user_dashboard_summary
user_progress
users

7. Once created, go to Project Settings → Data API

8. Copy your SUPABASE_KEY, then add them to your .env



service_role secret

**** **** **** ****

This key has the ability to bypass Row Level Security. Never share it publicly. If leaked, generate a new JWT secret immediately. Prefer using Secret API keys instead.

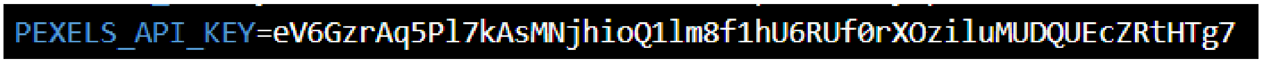
No requests in the past 24 hours.

- **Activity 3.4: Use Pexels API with PEXELS_API_KEY to add visual learning resources.**

  Implement Pexels API (PEXELS_API_KEY) to enrich the learning dashboard with contextual media resources, such as infographics and visual content for communication skill training.

  This activity integrates the Pexels API into the VidyāMitra platform to enhance the user learning experience with engaging visual content. Using the PEXELS_API_KEY, the backend connects to Pexels, a free stock media service, to fetch high-quality images, infographics, and visual aids related to communication, presentation, and professional skill development. These visuals are dynamically displayed within the user's training dashboard to make the learning journey more interactive and visually rich. This integration helps users improve soft skills through a more immersive and motivating interface.

https://www.pexels.com/api/new/



```
PEXELS_API_KEY=eV6GzrAq5Pl7kAsMNjhioQ1lm8f1hU6RUf0rXOziluMUDQUEcZRtHTg7
```

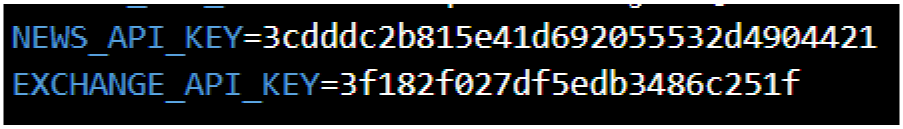
1. Sign in or create a free account on Pexels.
2. Visit the API Dashboard from your profile menu.
3. Click "Generate API Key."
4. Copy the key provided.
5. Add it to your .env

- **Activity 3.5: Fetch live market and financial updates via News and Exchange APIs.**

Integrate News and Exchange APIs (NEWS_API_KEY, EXCHANGE_API_KEY) to provide users with live updates on global market trends and financial data relevant to their chosen career domain.

This activity integrates the News and Exchange APIs into the VidyāMitra platform to deliver real-time insights on global trends and financial market updates. Using the NEWS_API_KEY and EXCHANGE_API_KEY, the backend fetches live data such as current events, job market changes, and currency exchange rates relevant to users' selected career domains. These updates are displayed on the user's dashboard, helping learners and professionals stay informed about industry movements and economic factors that influence their career paths. This integration adds a dynamic and practical dimension to VidyāMitra's AI-driven career guidance system.

News API: https://newsapi.org/
Exchange Rate API: https://www.exchangerate-api.com/



```
NEWS_API_KEY=3cdddc2b815e41d692055532d4904421
EXCHANGE_API_KEY=3f182f027df5edb3486c251f
```

1. Go to the respective website (News API or Exchange Rate API).
2. Sign up for a free developer account.
3. After verification, navigate to your API Dashboard.

4. Click “Generate API Key” to create your unique key.

5. Add them to your .env

Conclusion:

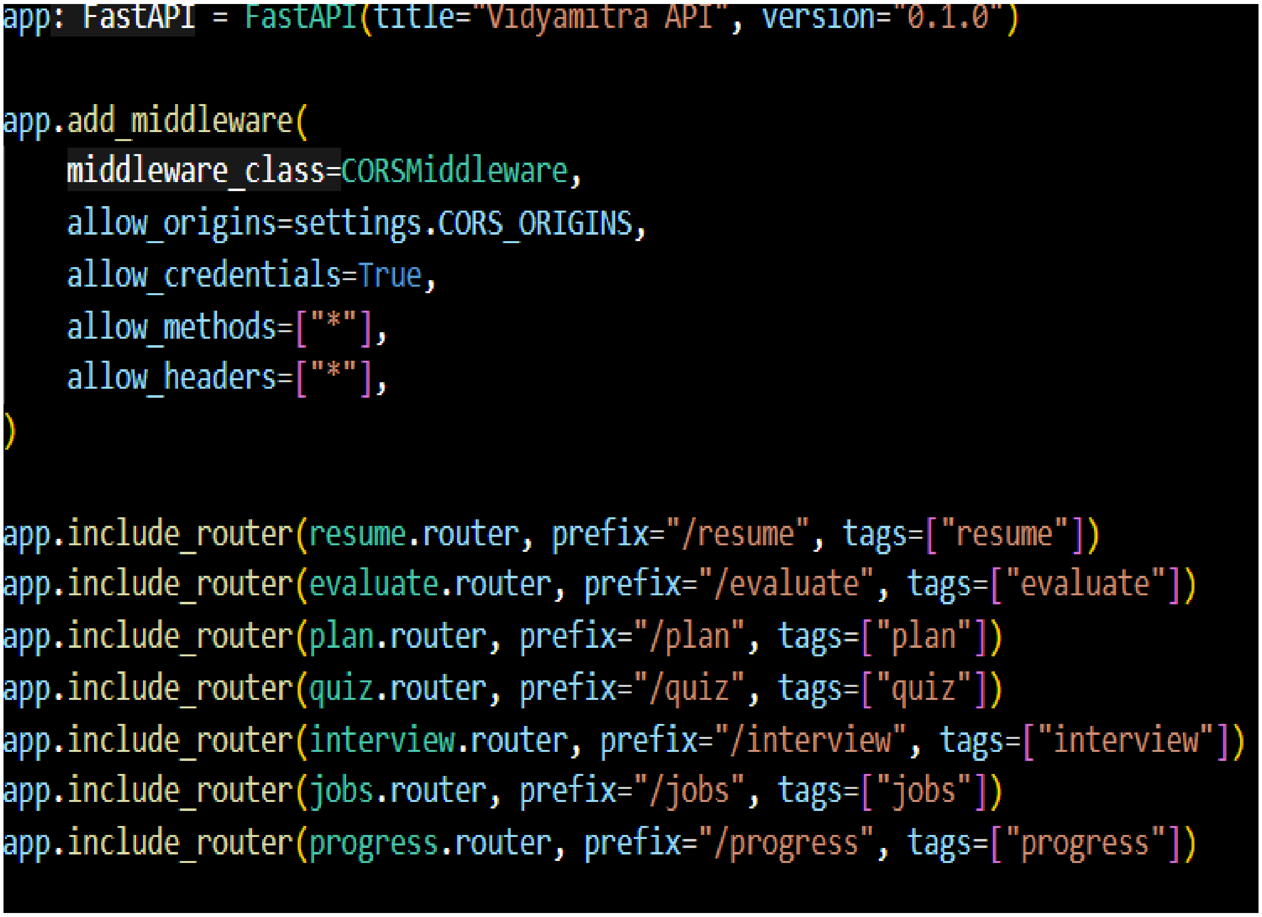
Milestone 3 focuses on integrating AI and external APIs into VidyāMitra to provide intelligent, personalized career guidance. It includes connecting the GPT-4 API for resume analysis and interview simulation, fetching tailored learning resources via Google and YouTube APIs, using Supabase for secure real-time data storage, adding visual content through the Pexels API, and delivering live market and financial updates via News and Exchange APIs. These integrations create a dynamic, interactive, and context-aware platform to enhance user learning and career development.

Milestone 4: React.js Frontend Development

- **Activity 4.1: Create a responsive dashboard for all career modules.**

  Develop a responsive career analytics dashboard for all user modules Resume Evaluation, Skill Mapping, Training Planner, Quiz Performance, and Job Recommendations.

  This section of the code sets up the Vidyamitra API using FastAPI and organizes the backend into multiple functional modules. It includes the CORS middleware to allow communication between the backend and the React frontend so that data can be securely shared across different origins. Each feature of the system such as resume, evaluate, plan, quiz, interview, jobs, and progress is defined as a separate router with its own URL prefix and tag. This modular approach keeps the backend organized and easy to manage, allowing each feature to work independently while still being part of the same application. Overall, this structure helps handle different API requests efficiently and ensures a smooth flow between the system's various components.



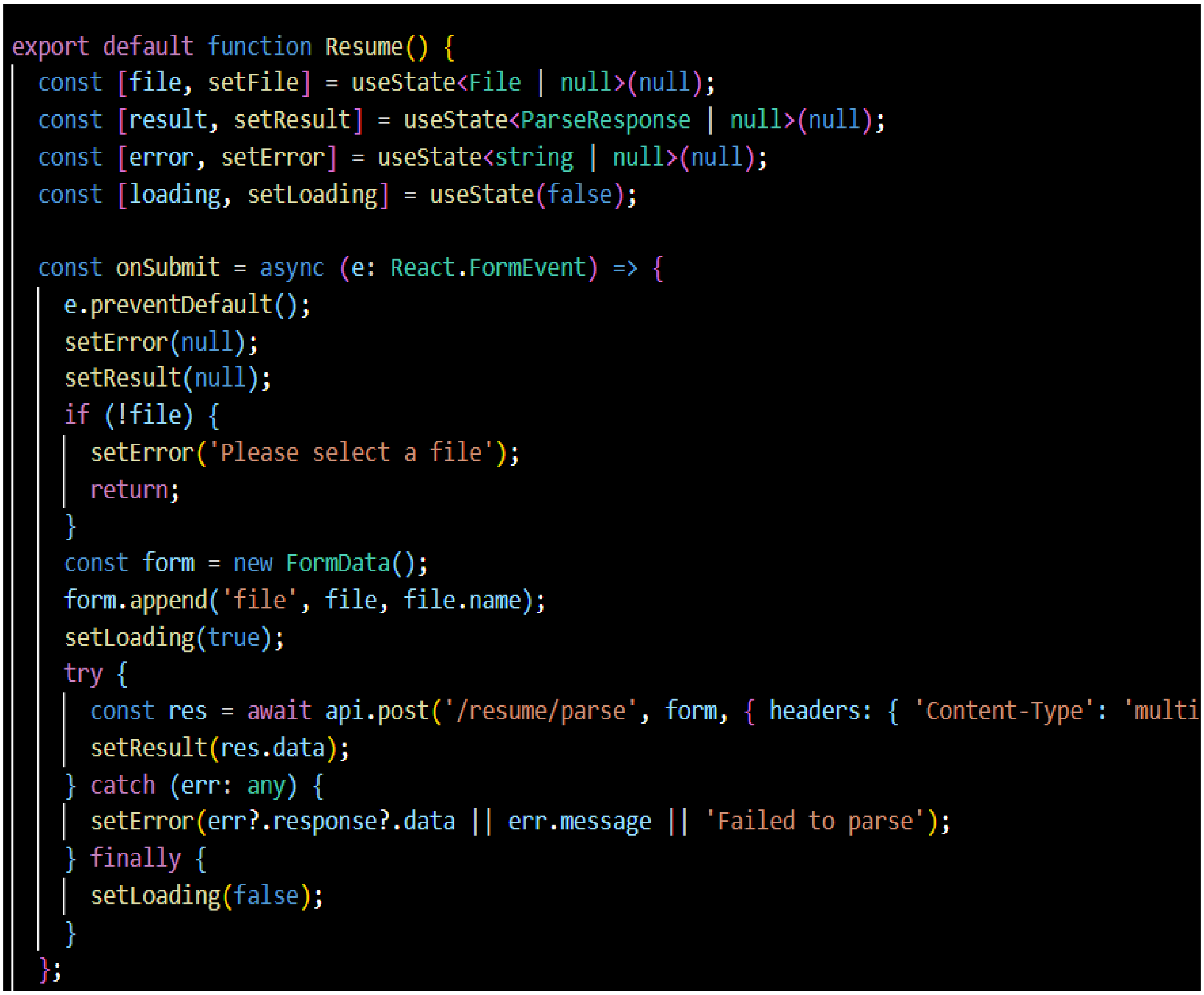
```
app: FastAPI = FastAPI(title="Vidyamitra API", version="0.1.0")

app.add_middleware(
    middleware_class=CORSMiddleware,
    allow_origins=settings.CORS_ORIGINS,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.include_router(resume.router, prefix="/resume", tags=["resume"])
app.include_router(evaluate.router, prefix="/evaluate", tags=["evaluate"])
app.include_router(plan.router, prefix="/plan", tags=["plan"])
app.include_router(quiz.router, prefix="/quiz", tags=["quiz"])
app.include_router(interview.router, prefix="/interview", tags=["interview"])
app.include_router(jobs.router, prefix="/jobs", tags=["jobs"])
app.include_router(progress.router, prefix="/progress", tags=["progress"])
```

- **Activity 4.2: Create interactive pages for resume, skills, training, and interviews.**

  Create interactive pages for resume upload, skill-gap visualization, training progress tracking, and mock interview feedback, ensuring seamless navigation across components.

  Resume Upload Functionality:

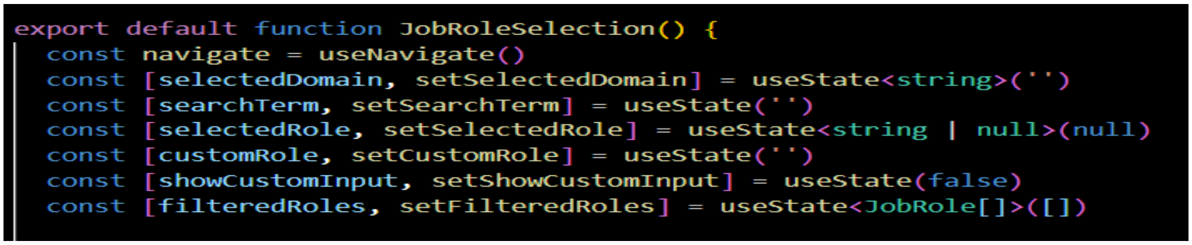
  This React component handles the resume upload feature in the Vidyamitra platform. It allows users to select a file, validates the input, and sends it to the backend API endpoint /resume/parse using a POST request. The component manages loading, error, and result states to provide feedback during the upload process. If the upload is successful, the parsed data is displayed to the user; otherwise, an error message appears. This ensures a smooth and responsive resume parsing experience for users.



```
export default function Resume() {
  const [file, setFile] = useState<File | null>(null);
  const [result, setResult] = useState<ParseResponse | null>(null);
  const [error, setError] = useState<string | null>(null);
  const [loading, setLoading] = useState(false);

  const onSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setError(null);
    setResult(null);
    if (!file) {
      setError('Please select a file');
      return;
    }
    const form = new FormData();
    form.append('file', file, file.name);
    setLoading(true);
    try {
      const res = await api.post('/resume/parse', form, { headers: { 'Content-Type': 'multi
      setResult(res.data);
    } catch (err: any) {
      setError(err?.response?.data || err.message || 'Failed to parse');
    } finally {
      setLoading(false);
    }
  };
```

Job Role Selection Module:

This component allows users to choose or search for their desired job roles within the Vidyamitra platform. It uses React state hooks to manage selected domains, search terms, and filtered job roles dynamically. Users can either pick from predefined options or create a custom job role if their desired one is not listed. The component updates results in real time based on user input, making role selection quick and interactive. Overall, it serves as the first step in personalizing the user's career path and training recommendations.



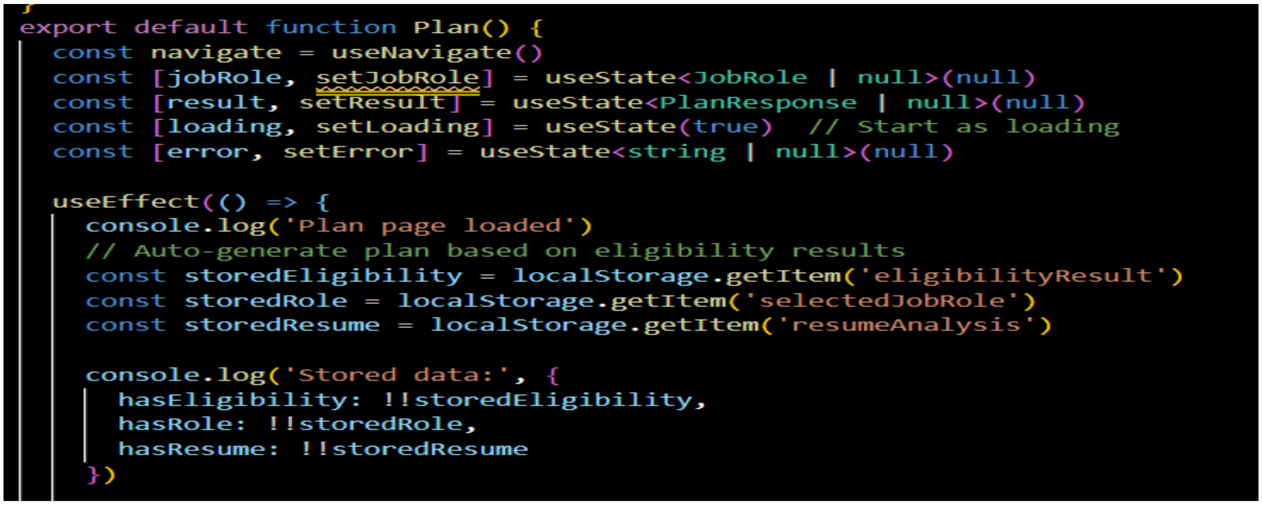
```
export default function JobRoleSelection() {
  const navigate = useNavigate()
  const [selectedDomain, setSelectedDomain] = useState<string>('')
  const [searchTerm, setSearchTerm] = useState('')
  const [selectedRole, setSelectedRole] = useState<string | null>(null)
  const [customRole, setCustomRole] = useState('')
  const [showCustomInput, setShowCustomInput] = useState(false)
  const [filteredRoles, setFilteredRoles] = useState<JobRole[]>([])
```

Training Plan Module:

This component handles the generation of personalized training plans for users. It retrieves stored data like eligibility results, selected job role, and resume details from local storage. Based on this information, it prepares the user's customized learning path. The page automatically loads the relevant data when opened and manages loading states. This helps users view and follow AI-generated training plans aligned with their career goals. This component manages the creation of personalized training plans for each user in the Vidyamitra platform. It retrieves eligibility results, selected job roles, and resume analysis data from local storage to generate targeted learning paths.

Using this data, it helps users identify skill gaps and provides structured recommendations for upskilling. The component automatically loads data when the page is opened and displays progress in real time.It also handles loading and error states to ensure a smooth and reliable user experience.

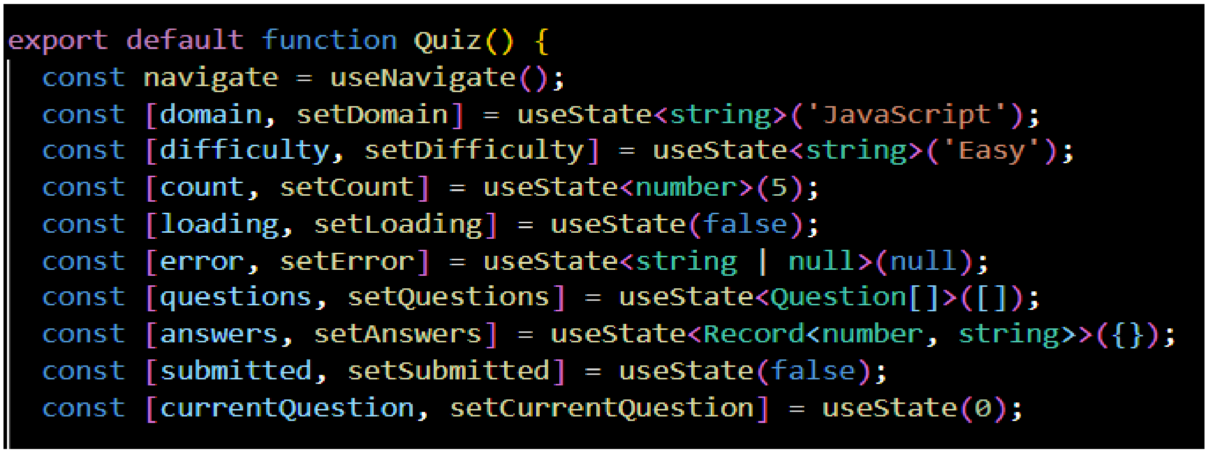
Overall, this module acts as a bridge between resume evaluation and personalized skill development guidance.



```
export default function Plan() {
  const navigate = useNavigate()
  const [jobRole, setJobRole] = useState<JobRole | null>(null)
  const [result, setResult] = useState<PlanResponse | null>(null)
  const [loading, setLoading] = useState(true)  // Start as loading
  const [error, setError] = useState<string | null>(null)

  useEffect(() => {
    console.log('Plan page loaded')
    // Auto-generate plan based on eligibility results
    const storedEligibility = localStorage.getItem('eligibilityResult')
    const storedRole = localStorage.getItem('selectedJobRole')
    const storedResume = localStorage.getItem('resumeAnalysis')

    console.log('Stored data:', {
      hasEligibility: !!storedEligibility,
      hasRole: !!storedRole,
      hasResume: !!storedResume
    })
```

Quiz Module:

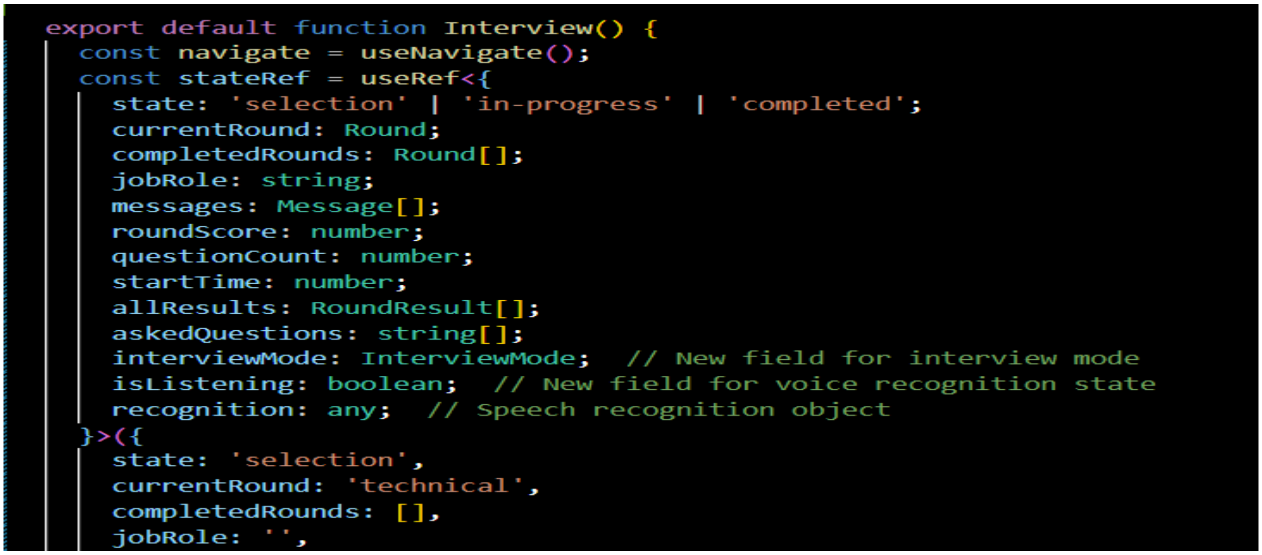
This component manages the quiz functionality within the Vidyamitra platform.It allows users to take skill-based quizzes by selecting a domain, difficulty level, and number of questions. React state hooks are used to handle quiz data such as questions, answers, progress, and submission status.The component dynamically updates the current question and tracks user responses for evaluation. Overall, it helps users assess their knowledge and measure learning progress in real time.



```
export default function Quiz() {
  const navigate = useNavigate();
  const [domain, setDomain] = useState<string>('JavaScript');
  const [difficulty, setDifficulty] = useState<string>('Easy');
  const [count, setCount] = useState<number>(5);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState<string | null>(null);
  const [questions, setQuestions] = useState<Question[]>([]);
  const [answers, setAnswers] = useState<Record<number, string>>({});
  const [submitted, setSubmitted] = useState(false);
  const [currentQuestion, setCurrentQuestion] = useState(0);
```

Mock Interview Module:

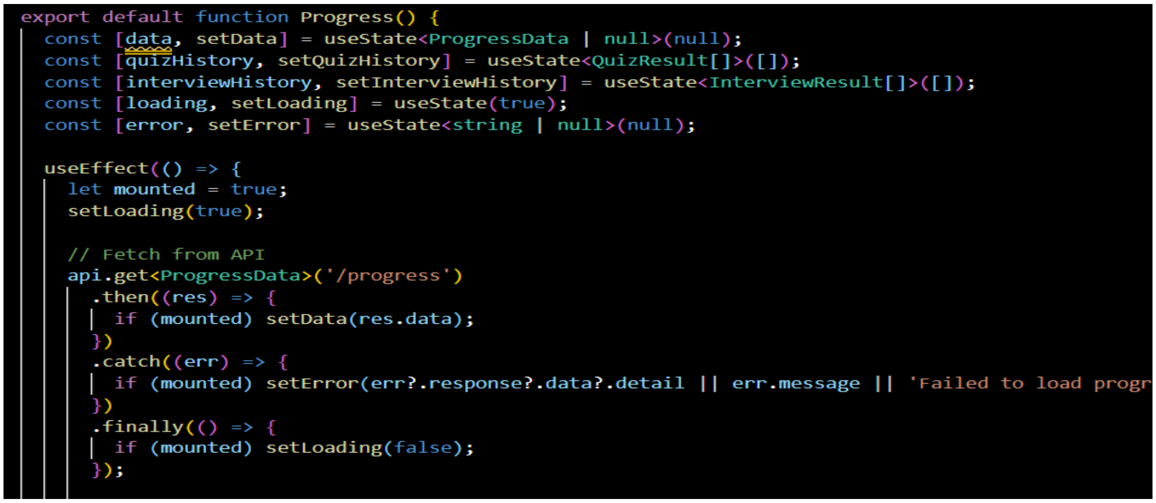
This component manages the mock interview functionality in the Vidyamitra platform. It tracks interview progress through different stages like selection, in-progress, and completion.The component stores details such as job role, score, questions asked, and interview rounds. It also supports voice-based interaction using speech recognition for a more realistic experience. Overall, it provides users with an AI-driven practice environment to enhance interview skills and confidence.



```
export default function Interview() {
  const navigate = useNavigate();
  const stateRef = useRef<{
    state: 'selection' | 'in-progress' | 'completed';
    currentRound: Round;
    completedRounds: Round[];
    jobRole: string;
    messages: Message[];
    roundScore: number;
    questionCount: number;
    startTime: number;
    allResults: RoundResult[];
    askedQuestions: string[];
    interviewMode: InterviewMode;  // New field for interview mode
    isListening: boolean;  // New field for voice recognition state
    recognition: any;  // Speech recognition object
  }>({
    state: 'selection',
    currentRound: 'technical',
    completedRounds: [],
    jobRole: '',
```

Progress Tracking Module:

This component is responsible for displaying user progress and performance data in the Vidyamitra platform. It fetches details like quiz history, interview results, and training progress from the backend API endpoint `/progress`. React state hooks manage loading, error, and data states to ensure smooth rendering and real-time updates. When the page loads, it retrieves the latest analytics to give users an overview of their learning journey. Overall, this module helps users track improvements and monitor their readiness across different career skills.



```
export default function Progress() {
  const [data, setData] = useState<ProgressData | null>(null);
  const [quizHistory, setQuizHistory] = useState<QuizResult[]>([]);
  const [interviewHistory, setInterviewHistory] = useState<InterviewResult[]>([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
    let mounted = true;
    setLoading(true);

    // Fetch from API
    api.get<ProgressData>('/progress')
      .then((res) => {
        if (mounted) setData(res.data);
      })
      .catch((err) => {
        if (mounted) setError(err?.response?.data?.detail || err.message || 'Failed to load progr
      })
      .finally(() => {
        if (mounted) setLoading(false);
      });
```

Conclusion:

Milestone 4 focuses on developing the React.js frontend for VidyāMitra, creating a responsive and interactive dashboard for all career modules. It includes pages for resume upload, skill-gap visualization, personalized training plans, quizzes, mock interviews, job role selection, and progress tracking, ensuring seamless navigation and real-time updates. This frontend development enables an engaging, user-friendly interface that connects smoothly with the backend and AI-driven career guidance features.

Milestone 5: Testing and Deployment

- ## Activity 5.1: Backend Testing and API Validation

  The backend, built using FastAPI, is tested to ensure all API endpoints function as expected. Authentication, color analysis, trend detection, and design generation routes are validated using sample inputs through Postman. Each API response is checked for accuracy, response time, and correct data structure. The Groq-powered reasoning and Hugging Face image generation APIs are tested to confirm that the backend correctly processes requests and returns meaningful outputs, such as generated fashion sketches and descriptive recommendations.

- ## Activity 5.2: Frontend Functionality Testing

  The React.js frontend is tested to verify smooth user interactions and accurate data presentation. Each page including Design Generator, Trend Analysis, and Color Analyzer is reviewed for responsiveness and proper rendering of results fetched from backend APIs. The user interface components, such as image upload panels, loading animations, and result cards, are tested to ensure a seamless and interactive user experience. Real-time updates and progress indicators are validated for proper functionality.

- ## Activity 5.3: Backend–Frontend Integration Testing

  Integration testing ensures that the FastAPI backend and React frontend communicate effectively through axios API calls. Actions such as user login, image upload, and AI design generation are executed to verify smooth data exchange. The frontend sends HTTP requests to the backend endpoints, and the responses including generated images, color palettes, and trend predictions are displayed instantly on the interface. JWT authentication is confirmed for secure and consistent request handling throughout the workflow.

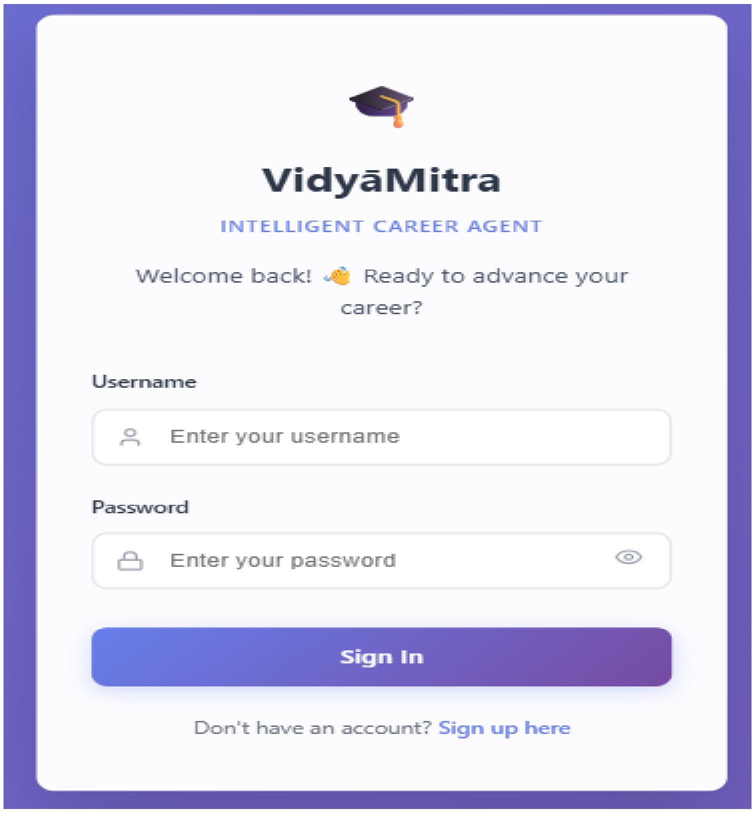
- ## Activity 5.4: Output Validation and User Experience Review

  The complete workflow is tested end-to-end to verify that the outputs are accurate and visually appealing. When users upload an image or provide a design prompt, the backend AI modules process the input and return color extraction results, fashion recommendations, and AI-generated design previews. These outputs are displayed on the frontend in an organized and visually rich layout. The final user experience demonstrates seamless coordination between backend intelligence and frontend presentation, confirming that the system performs efficiently in a local environment.

Conclusion:

Milestone 5 covers testing and deployment of the VidyāMitra platform. The backend APIs and frontend components are thoroughly tested for functionality, responsiveness, and accurate data exchange. End-to-end integration ensures seamless communication, reliable outputs, and a smooth user experience.

Login page:



VidyāMitra

INTELLIGENT CAREER AGENT

Welcome back! 👋 Ready to advance your career?

Username

Enter your username

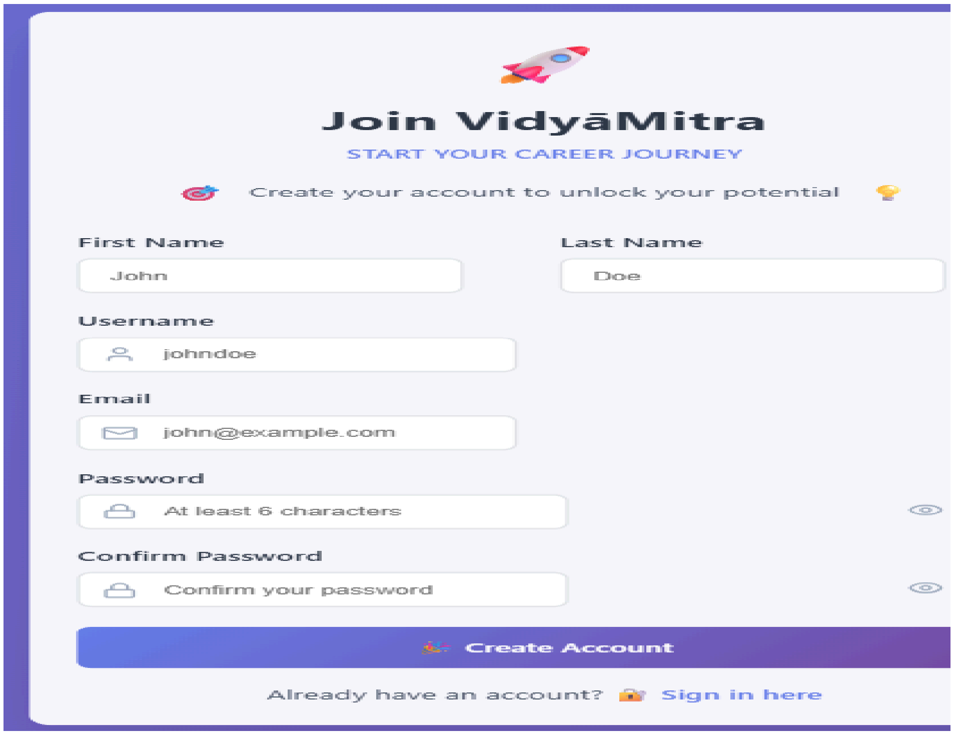
Password

Enter your password

Sign In

Don't have an account? Sign up here

Register page:



Join VidyāMitra

START YOUR CAREER JOURNEY

🎯 Create your account to unlock your potential 💡

First Name

John

Last Name

Doe

Username

johndoe

Email

john@example.com

Password

At least 6 characters

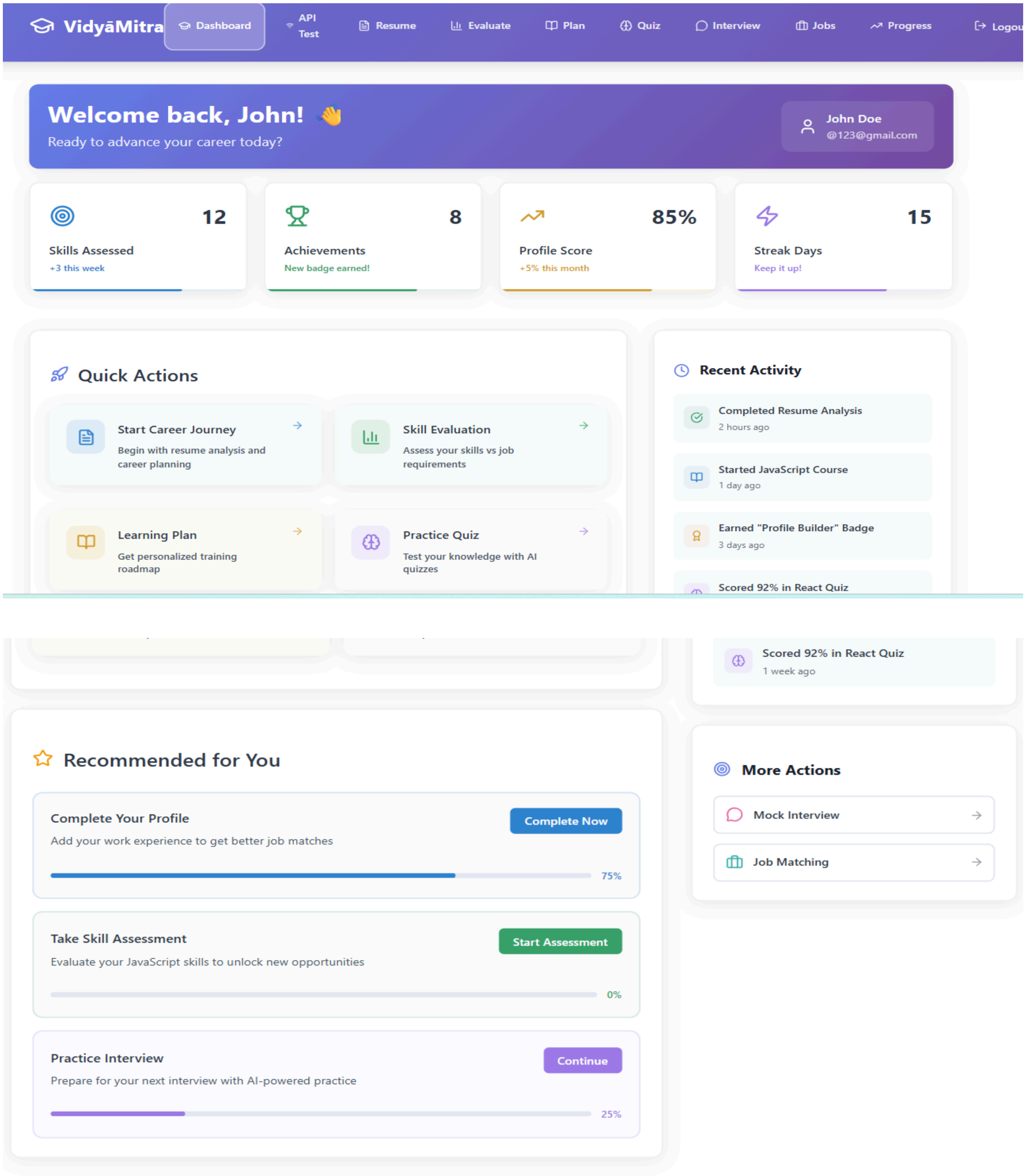
Confirm Password

Confirm your password

Create Account

Already have an account? Sign in here

Home page:



Welcome back, John! 👋

Ready to advance your career today?

John Doe
@123@gmail.com

12 Skills Assessed
+3 this week

8 Achievements
New badge earned!

85% Profile Score
+5% this month

15 Streak Days
Keep it up!

Quick Actions

Start Career Journey
Begin with resume analysis and career planning

Skill Evaluation
Assess your skills vs job requirements

Learning Plan
Get personalized training roadmap

Practice Quiz
Test your knowledge with AI quizzes

Recent Activity

Completed Resume Analysis
2 hours ago

Started JavaScript Course
1 day ago

Earned "Profile Builder" Badge
3 days ago

Scored 92% in React Quiz
1 week ago

Recommended for You

Complete Your Profile
Add your work experience to get better job matches
Complete Now
75%

Take Skill Assessment
Evaluate your JavaScript skills to unlock new opportunities
Start Assessment
0%

Practice Interview
Prepare for your next interview with AI-powered practice
Continue
25%

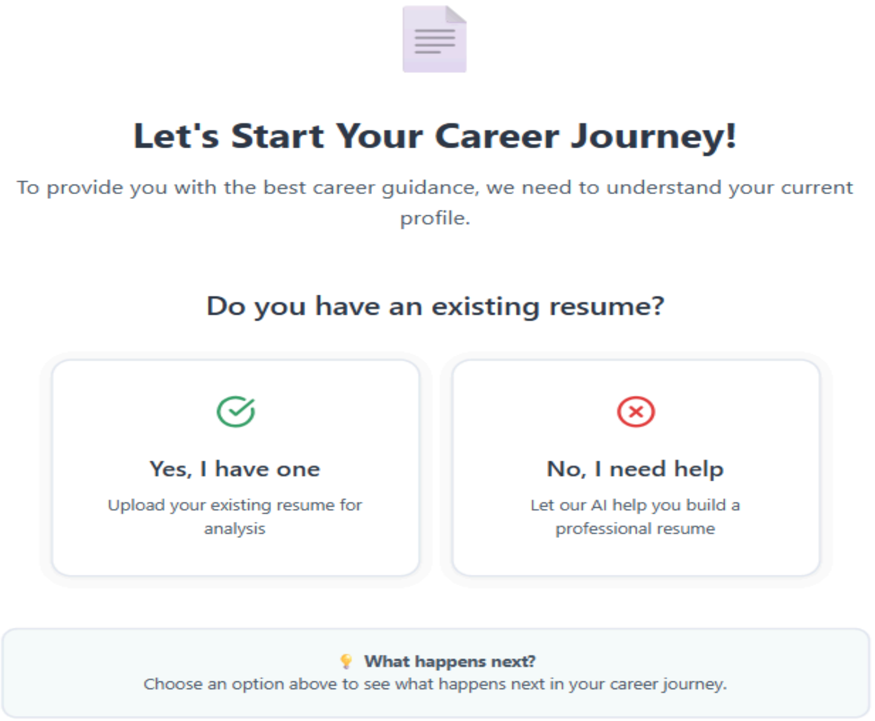
More Actions

Mock Interview

Job Matching

Start Your Career Journey page:

If User Has Resume:



Let's Start Your Career Journey!

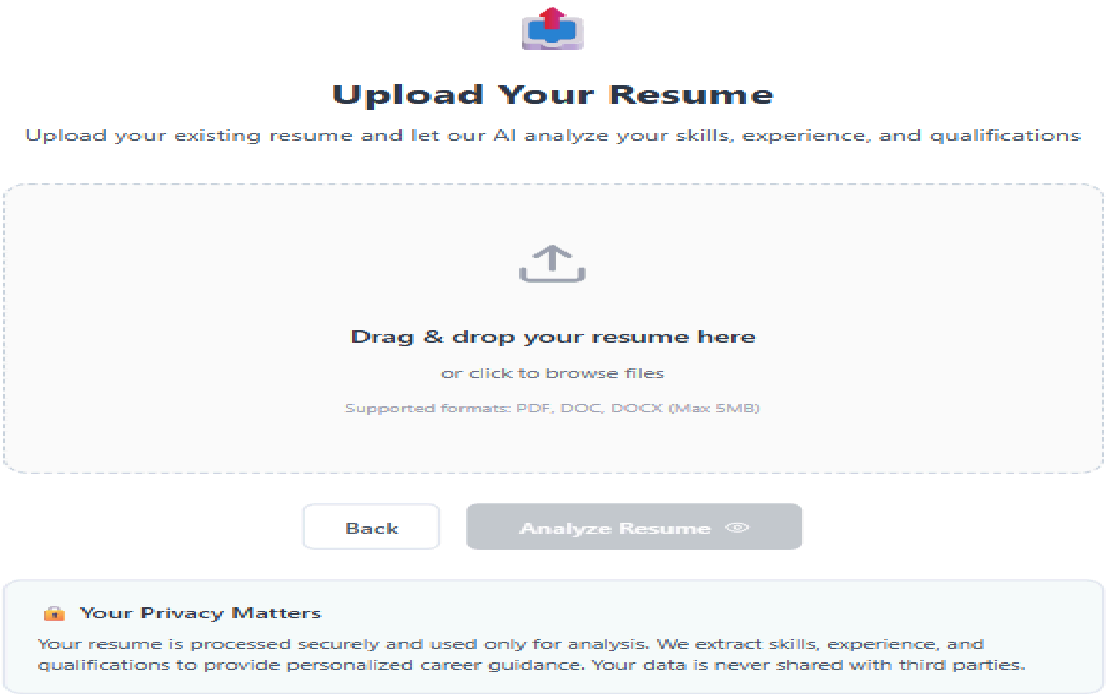
To provide you with the best career guidance, we need to understand your current profile.

Do you have an existing resume?

Yes, I have one
Upload your existing resume for analysis

No, I need help
Let our AI help you build a professional resume

💡 **What happens next?**
Choose an option above to see what happens next in your career journey.



Upload Your Resume

Upload your existing resume and let our AI analyze your skills, experience, and qualifications

Drag & drop your resume here

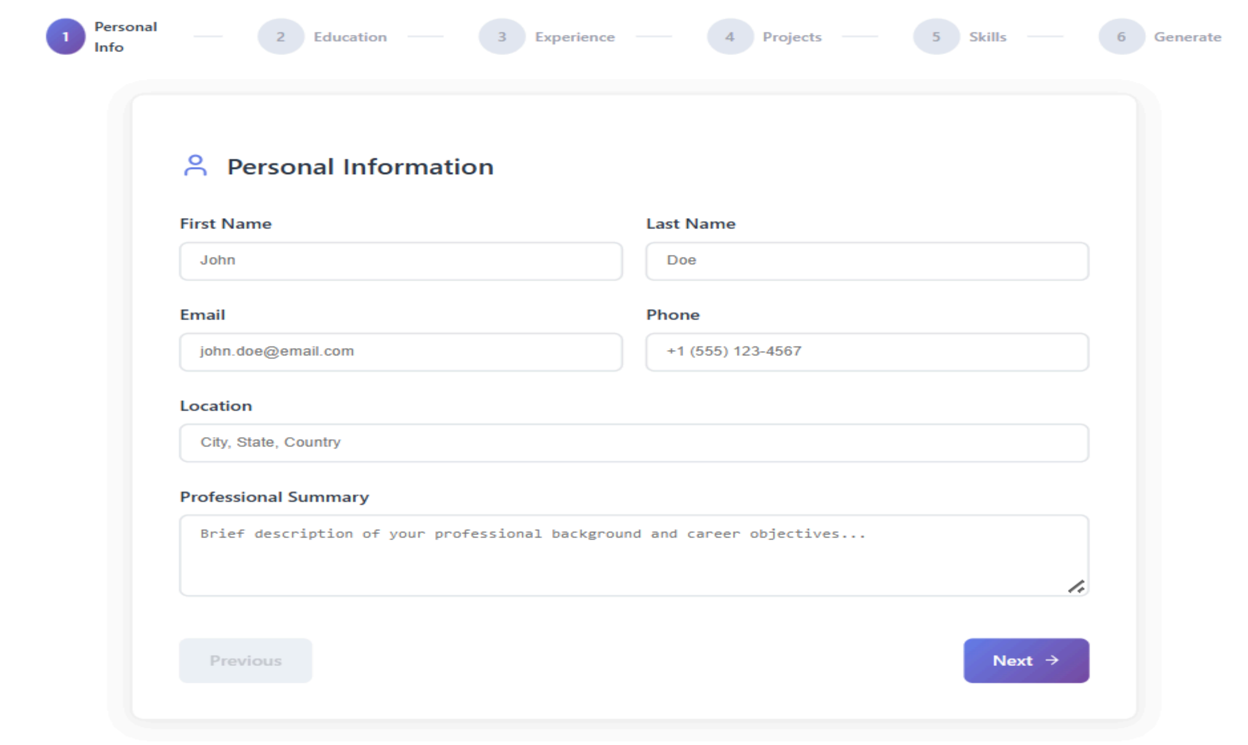
or click to browse files

Supported formats: PDF, DOC, DOCX (Max 5MB)

🔒 **Your Privacy Matters**

Your resume is processed securely and used only for analysis. We extract skills, experience, and qualifications to provide personalized career guidance. Your data is never shared with third parties.

If User Doesn’t have resume:



Personal Information

First Name: John

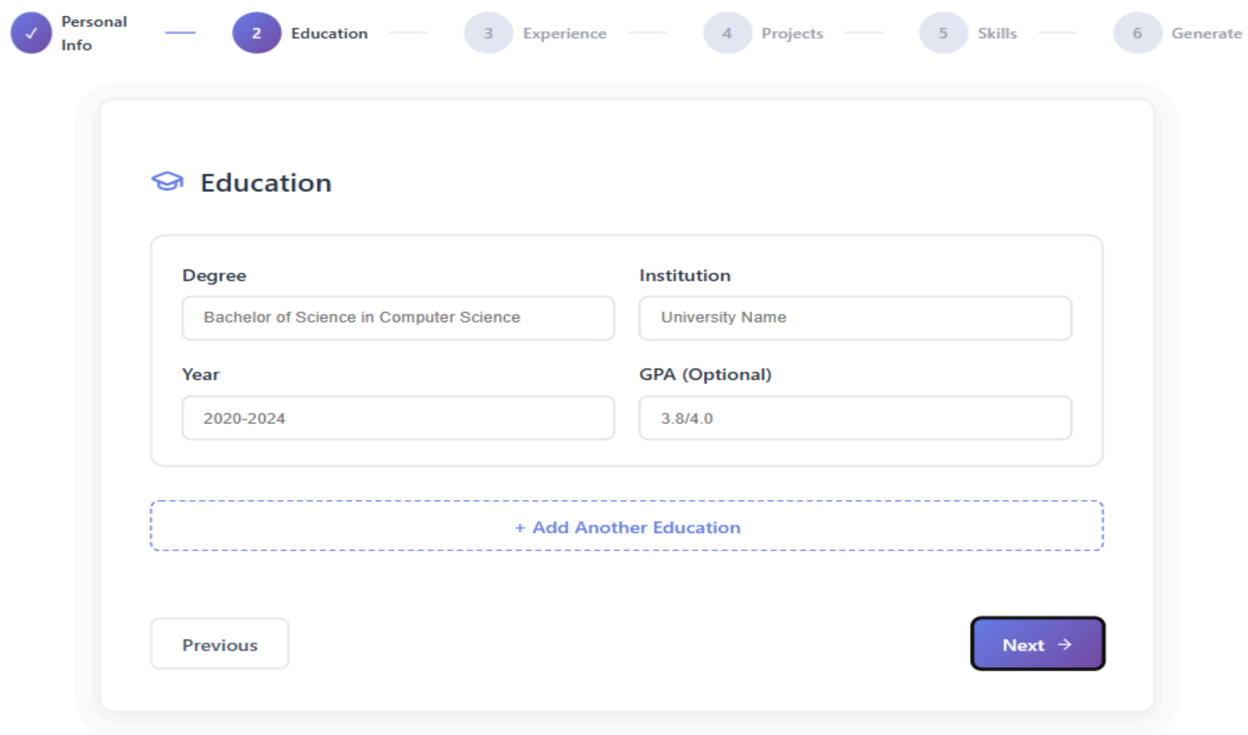
Last Name: Doe

Email: john.doe@email.com

Phone: +1 (555) 123-4567

Location: City, State, Country

Professional Summary: Brief description of your professional background and career objectives...



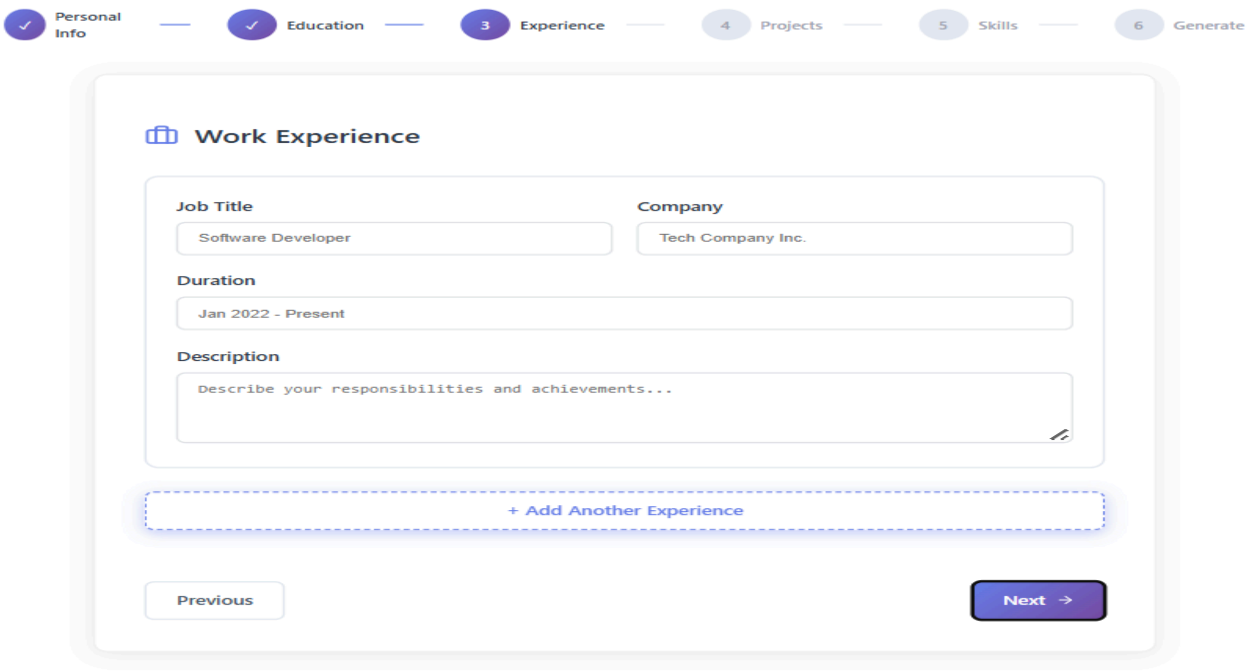
Education

Degree: Bachelor of Science in Computer Science

Institution: University Name

Year: 2020-2024

GPA (Optional): 3.8/4.0



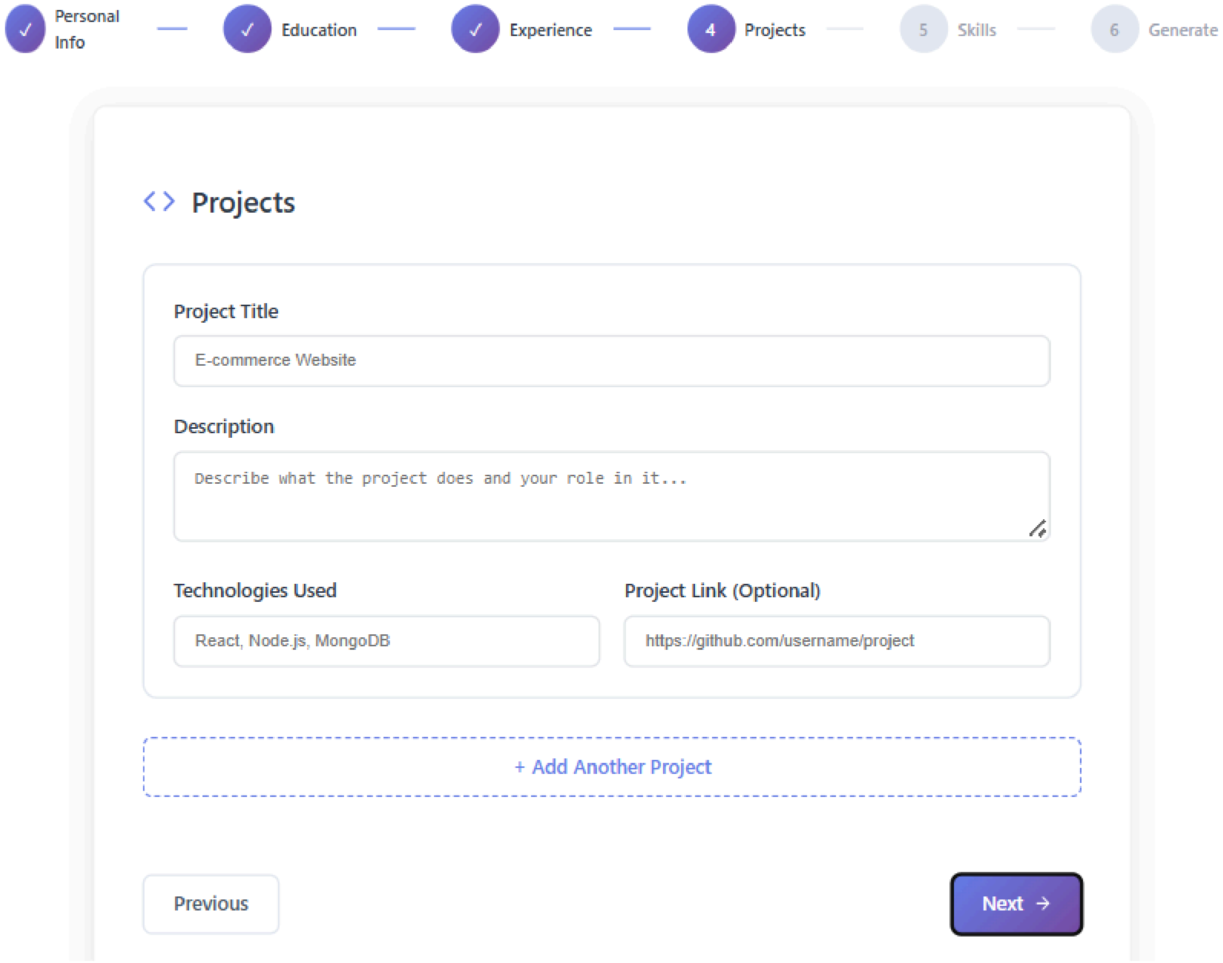
Work Experience

Job Title: Software Developer

Company: Tech Company Inc.

Duration: Jan 2022 - Present

Description: Describe your responsibilities and achievements...



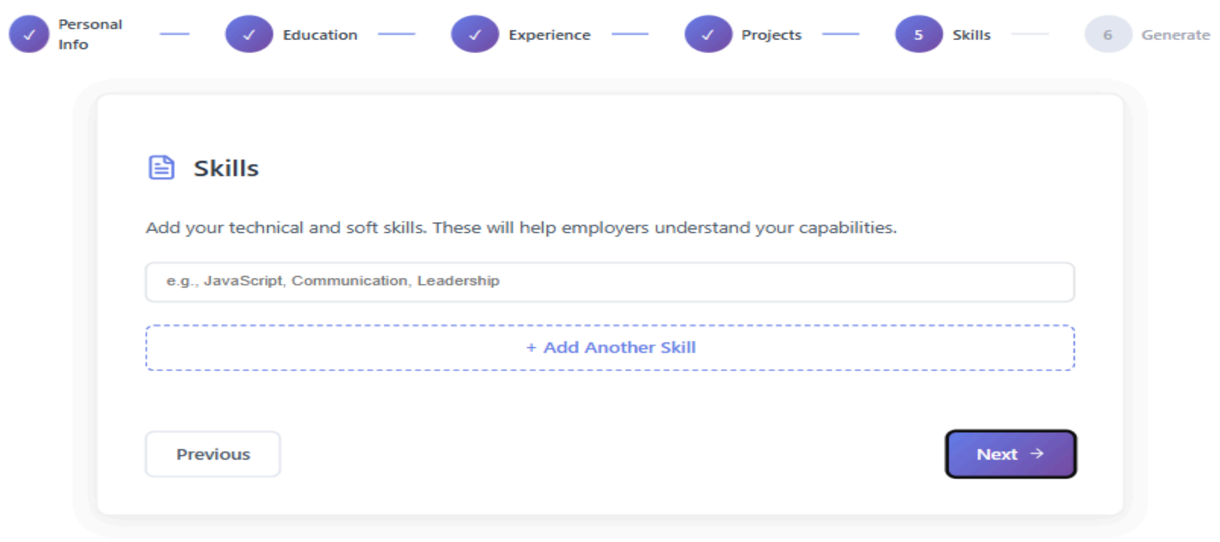
Projects

Project Title: E-commerce Website

Description: Describe what the project does and your role in it...

Technologies Used: React, Node.js, MongoDB

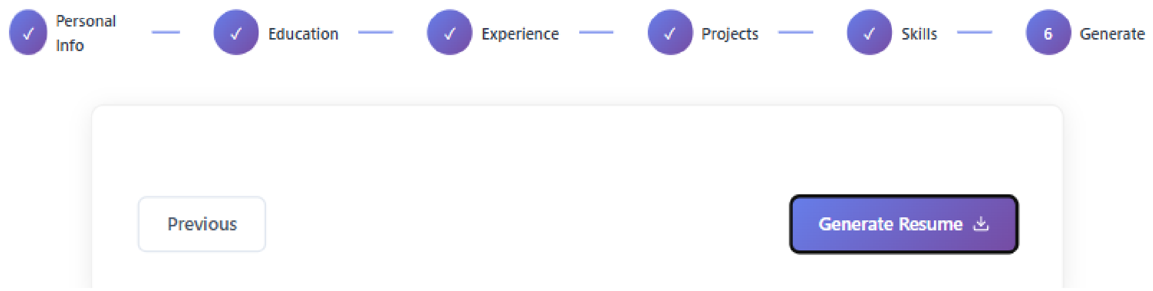
Project Link (Optional): https://github.com/username/project



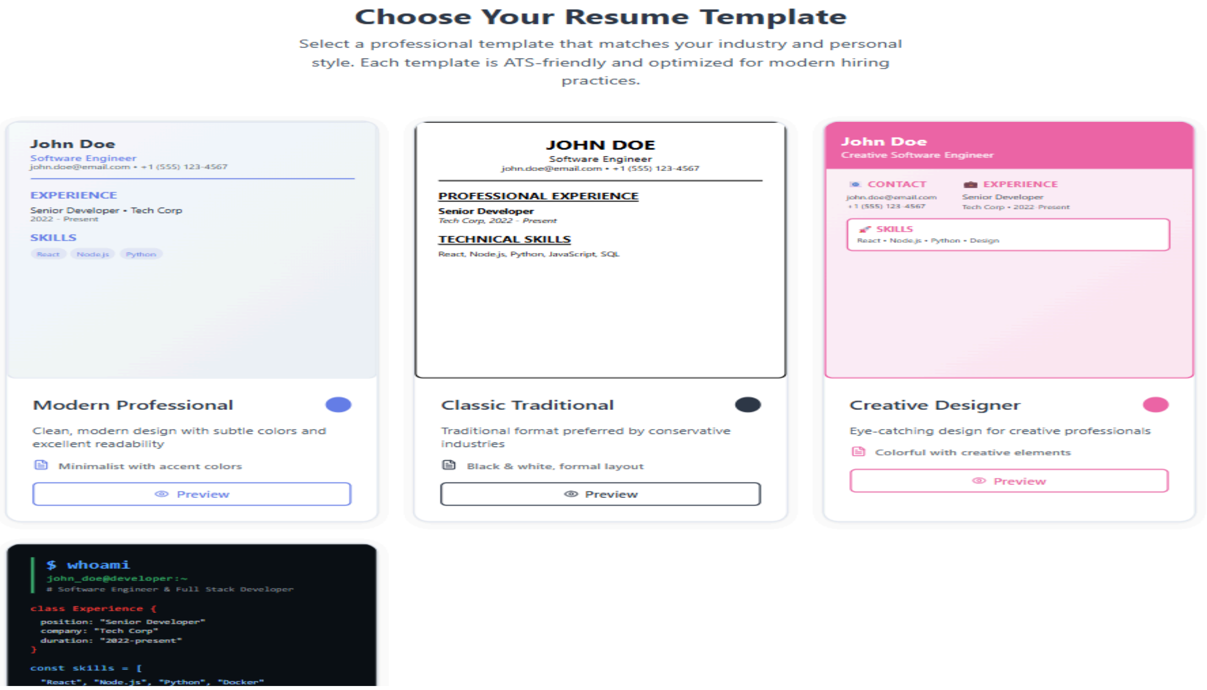
Skills

Add your technical and soft skills. These will help employers understand your capabilities.

e.g., JavaScript, Communication, Leadership



Resume Template Selection:



Choose Your Resume Template

Select a professional template that matches your industry and personal style. Each template is ATS-friendly and optimized for modern hiring practices.

John Doe
Software Engineer
john.doe@email.com • +1 (555) 123-4567
EXPERIENCE
Senior Developer • Tech Corp
2022 - Present
SKILLS
React Node.js Python

Modern Professional ●

Clean, modern design with subtle colors and excellent readability

Minimalist with accent colors

JOHN DOE
Software Engineer
john.doe@email.com • +1 (555) 123-4567
PROFESSIONAL EXPERIENCE
Senior Developer
Tech Corp, 2022 - Present
TECHNICAL SKILLS
React, Node.js, Python, JavaScript, SQL

Classic Traditional ●

Traditional format preferred by conservative industries

Black & white, formal layout

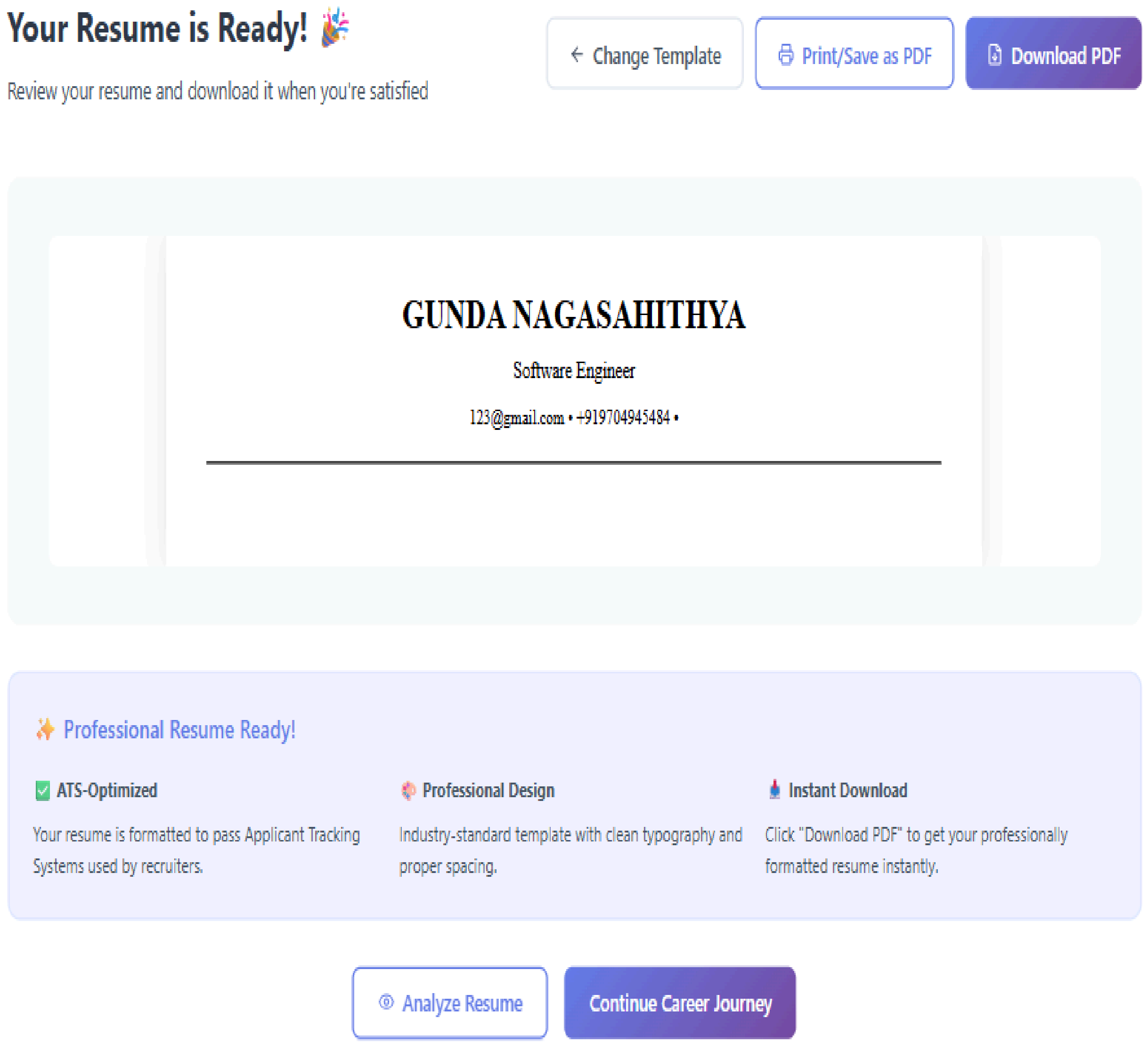
John Doe
Creative Software Engineer
CONTACT
john.doe@email.com
+1 (555) 123-4567
EXPERIENCE
Senior Developer
Tech Corp • 2022-Present
SKILLS
React • Node.js • Python • Design

Creative Designer ●

Eye-catching design for creative professionals

Colorful with creative elements

```
$ whoami
john_doe@developer:~
# Software Engineer & Full Stack Developer
class Experience {
  position: "Senior Developer"
  company: "Tech Corp"
  duration: "2022-present"
}
const skills = [
  "React", "Node.js", "Python", "Docker"
```



Your Resume is Ready! 🎉

Review your resume and download it when you're satisfied

GUNDA NAGASAHITHYA

Software Engineer

123@gmail.com • +919704945484 •

✨ Professional Resume Ready!

✅ **ATS-Optimized**

Your resume is formatted to pass Applicant Tracking Systems used by recruiters.

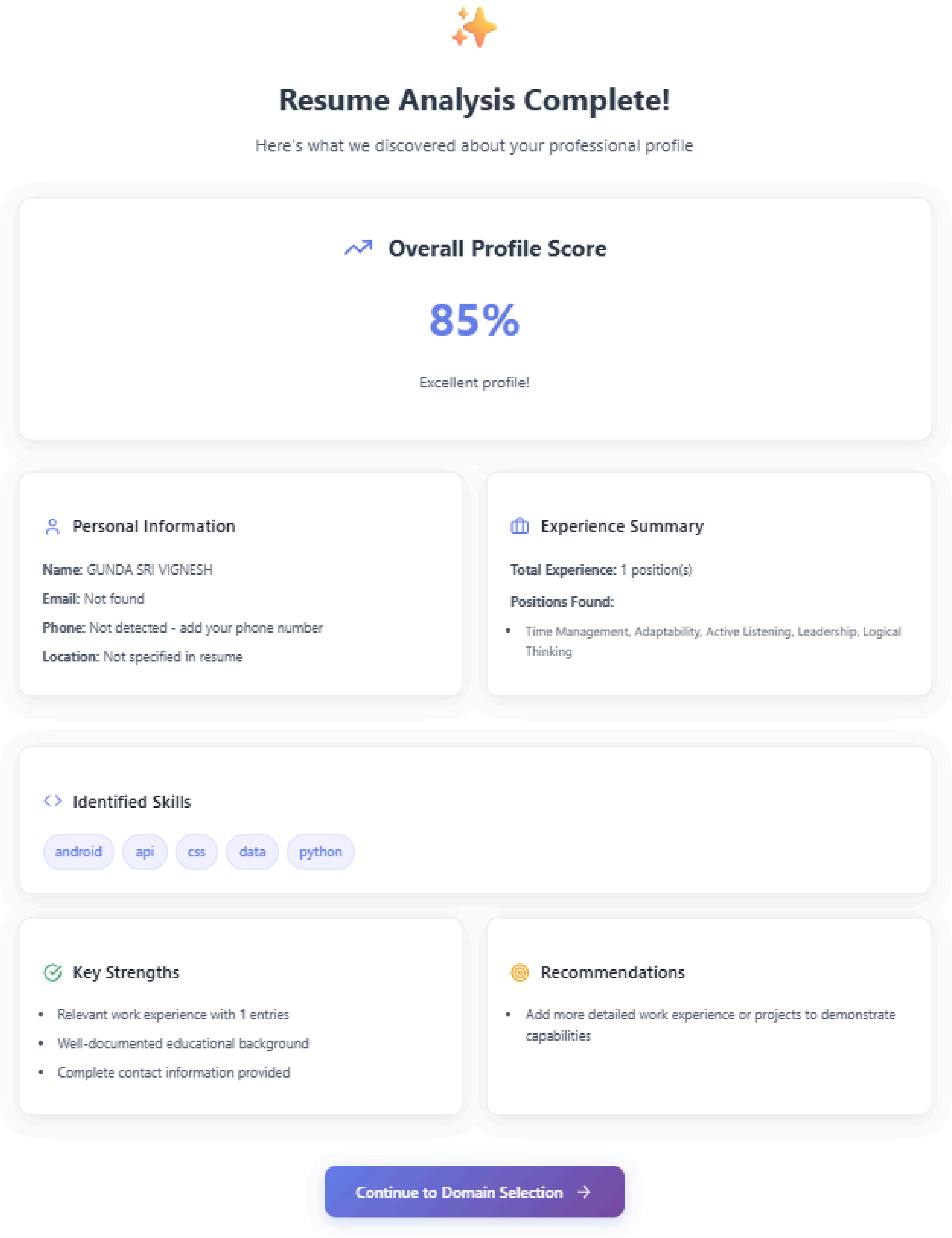
🎨 **Professional Design**

Industry-standard template with clean typography and proper spacing.

📥 **Instant Download**

Click "Download PDF" to get your professionally formatted resume instantly.

Resume Analysis Page:



Resume Analysis Complete!

Here's what we discovered about your professional profile

Overall Profile Score

85%

Excellent profile!

Personal Information

Name: GUNDA SRI VIGNESH

Email: Not found

Phone: Not detected - add your phone number

Location: Not specified in resume

Experience Summary

Total Experience: 1 position(s)

Positions Found:

- Time Management, Adaptability, Active Listening, Leadership, Logical Thinking

Identified Skills

android api css data python

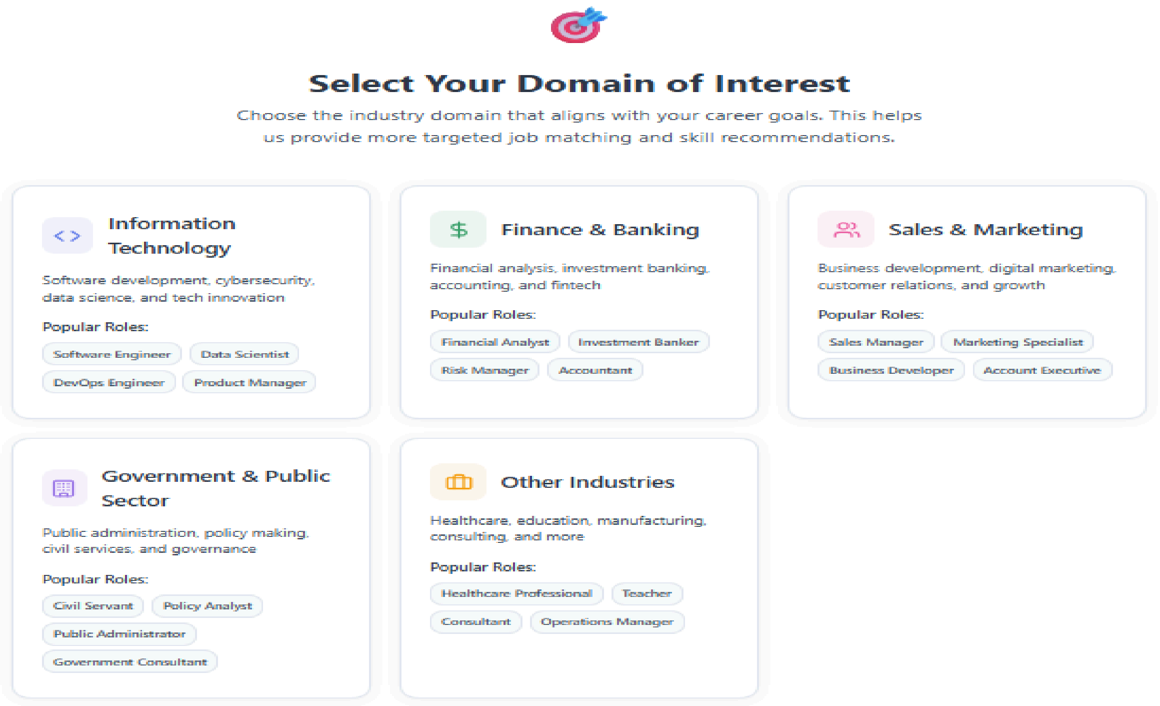
Key Strengths

- Relevant work experience with 1 entries
- Well-documented educational background
- Complete contact information provided

Recommendations

- Add more detailed work experience or projects to demonstrate capabilities

Domain Selection page:



Select Your Domain of Interest

Choose the industry domain that aligns with your career goals. This helps us provide more targeted job matching and skill recommendations.

Information Technology

Software development, cybersecurity, data science, and tech innovation

Popular Roles:

Software Engineer | Data Scientist | DevOps Engineer | Product Manager

Finance & Banking

Financial analysis, investment banking, accounting, and fintech

Popular Roles:

Financial Analyst | Investment Banker | Risk Manager | Accountant

Sales & Marketing

Business development, digital marketing, customer relations, and growth

Popular Roles:

Sales Manager | Marketing Specialist | Business Developer | Account Executive

Government & Public Sector

Public administration, policy making, civil services, and governance

Popular Roles:

Civil Servant | Policy Analyst | Public Administrator | Government Consultant

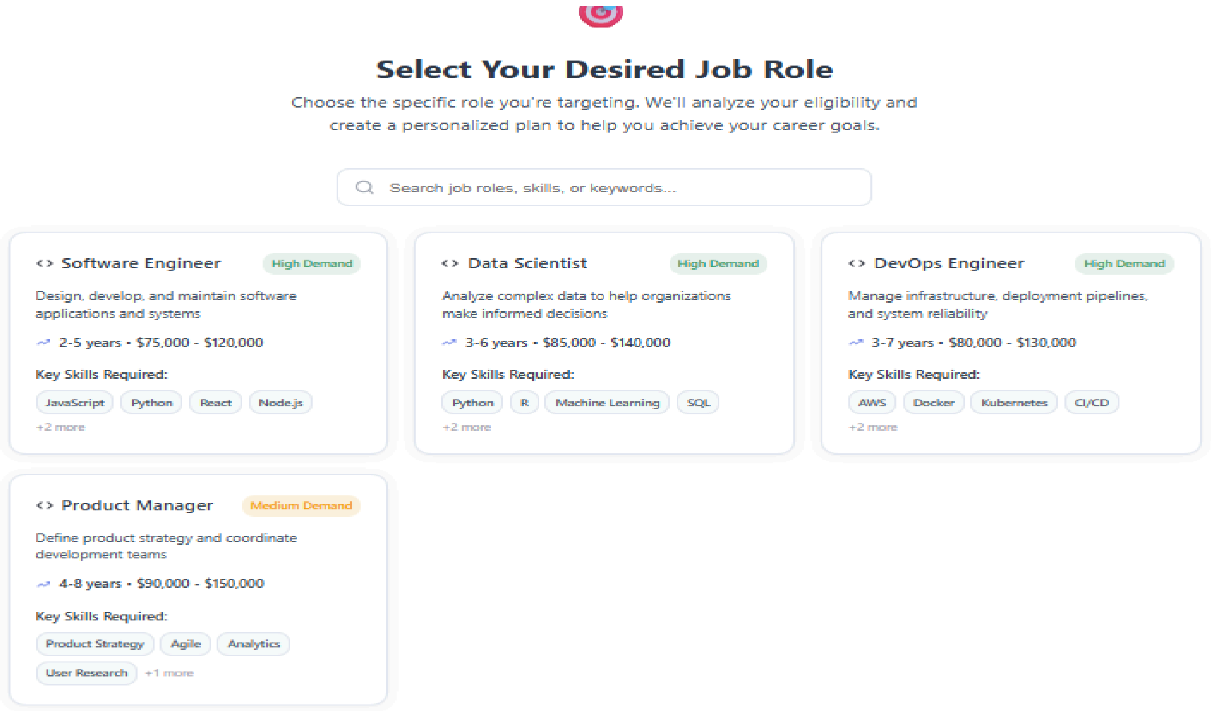
Other Industries

Healthcare, education, manufacturing, consulting, and more

Popular Roles:

Healthcare Professional | Teacher | Consultant | Operations Manager

Job Role Selection page:



Select Your Desired Job Role

Choose the specific role you're targeting. We'll analyze your eligibility and create a personalized plan to help you achieve your career goals.

<> Software Engineer — High Demand

Design, develop, and maintain software applications and systems

2-5 years • $75,000 - $120,000

Key Skills Required:

JavaScript | Python | React | Node.js

+2 more

<> Data Scientist — High Demand

Analyze complex data to help organizations make informed decisions

3-6 years • $85,000 - $140,000

Key Skills Required:

Python | R | Machine Learning | SQL

+2 more

<> DevOps Engineer — High Demand

Manage infrastructure, deployment pipelines, and system reliability

3-7 years • $80,000 - $130,000

Key Skills Required:

AWS | Docker | Kubernetes | CI/CD

+2 more

<> Product Manager — Medium Demand

Define product strategy and coordinate development teams

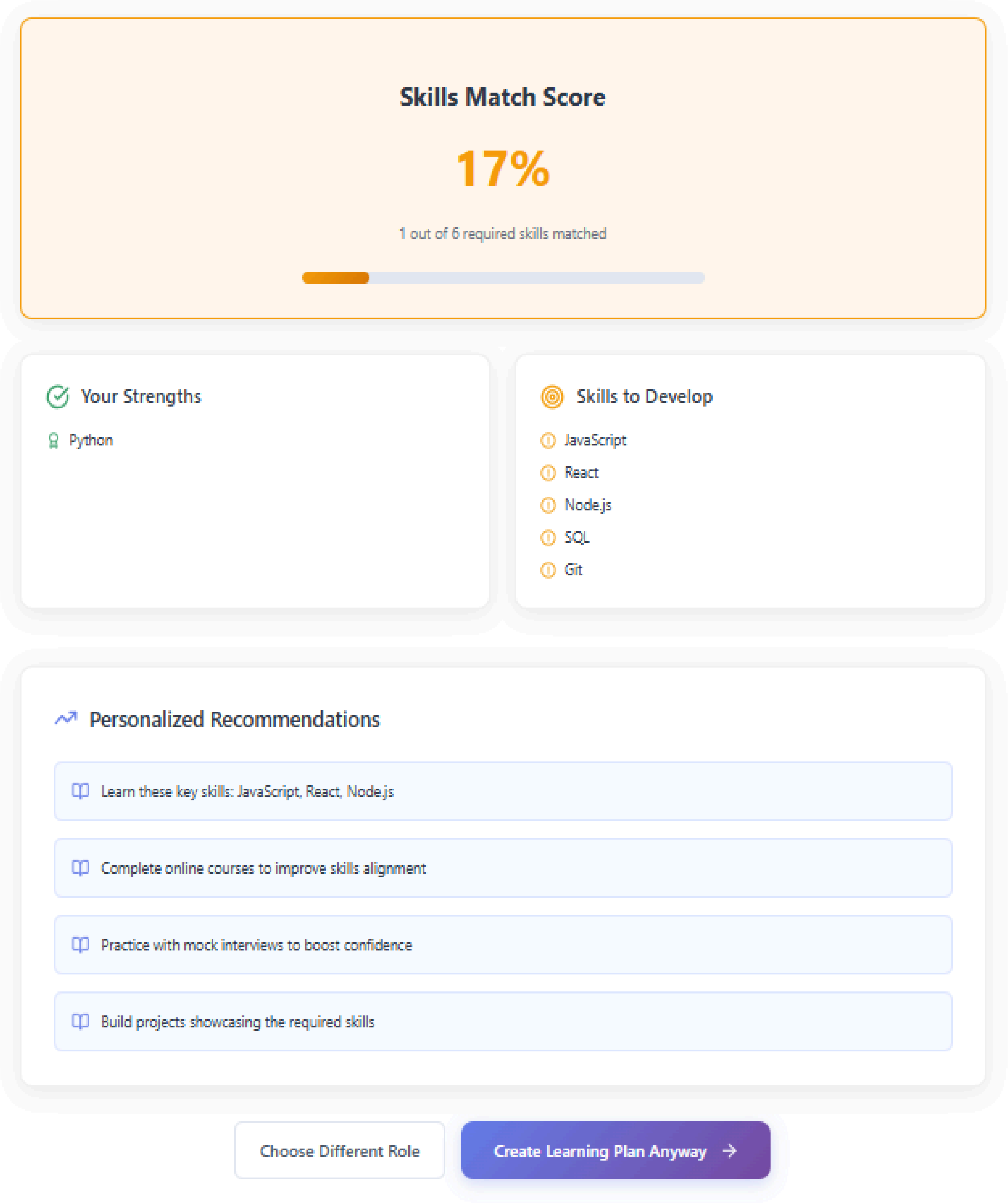
4-8 years • $90,000 - $150,000

Key Skills Required:

Product Strategy | Agile | Analytics | User Research

+1 more

Eligibility Check Page:



Skills Match Score

17%

1 out of 6 required skills matched

Your Strengths

- Python

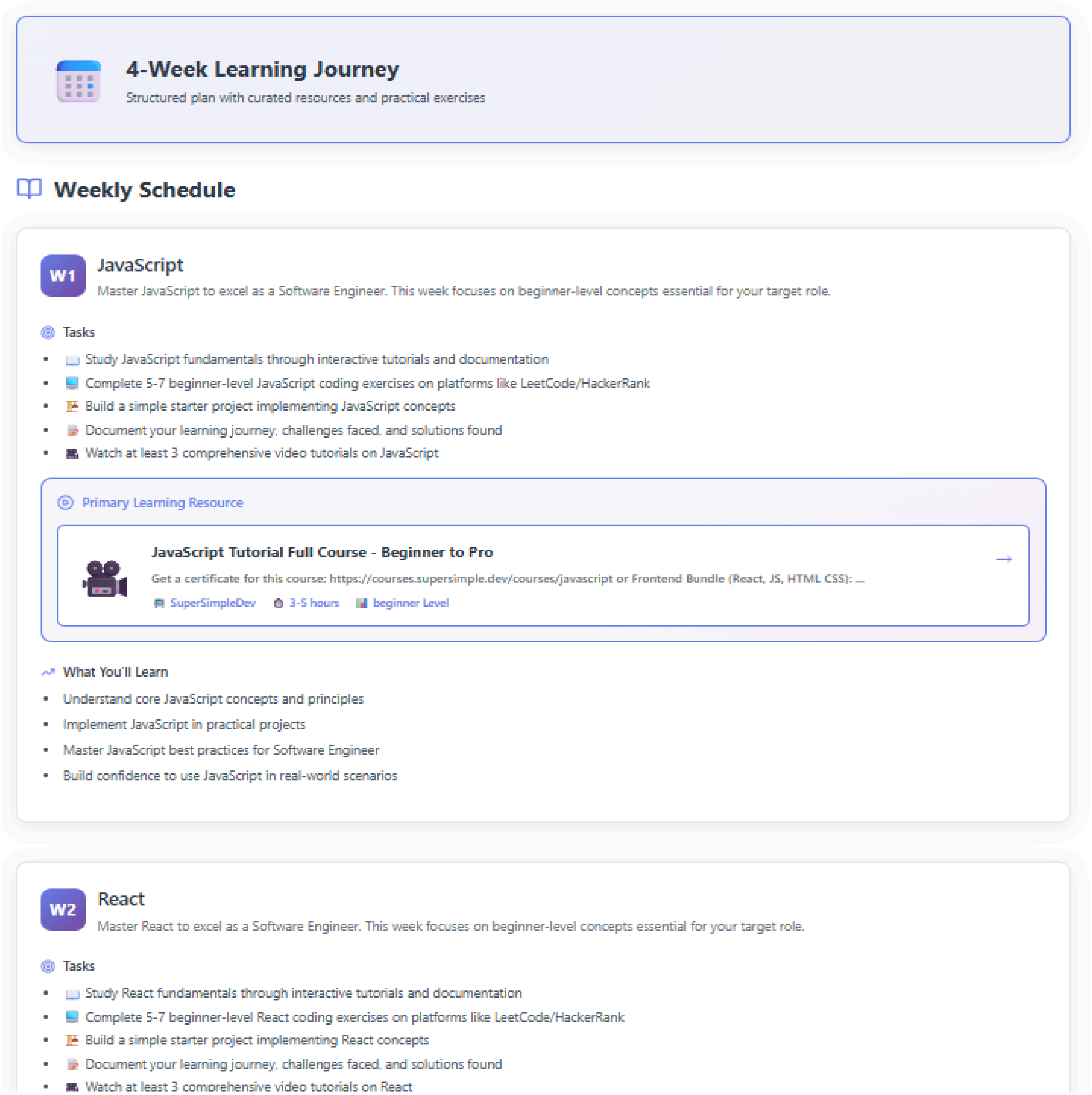
Skills to Develop

- JavaScript
- React
- Node.js
- SQL
- Git

Personalized Recommendations

- Learn these key skills: JavaScript, React, Node.js
- Complete online courses to improve skills alignment
- Practice with mock interviews to boost confidence
- Build projects showcasing the required skills

Learning Plan Page:



4-Week Learning Journey

Structured plan with curated resources and practical exercises

Weekly Schedule

W1 **JavaScript**

Master JavaScript to excel as a Software Engineer. This week focuses on beginner-level concepts essential for your target role.

Tasks

- 📖 Study JavaScript fundamentals through interactive tutorials and documentation
- 💻 Complete 5-7 beginner-level JavaScript coding exercises on platforms like LeetCode/HackerRank
- 🏗 Build a simple starter project implementing JavaScript concepts
- 📝 Document your learning journey, challenges faced, and solutions found
- 🎥 Watch at least 3 comprehensive video tutorials on JavaScript

Primary Learning Resource

JavaScript Tutorial Full Course - Beginner to Pro

Get a certificate for this course: https://courses.supersimple.dev/courses/javascript or Frontend Bundle (React, JS, HTML CSS): ...

SuperSimpleDev ⏱ 3-5 hours 📊 beginner Level

What You'll Learn

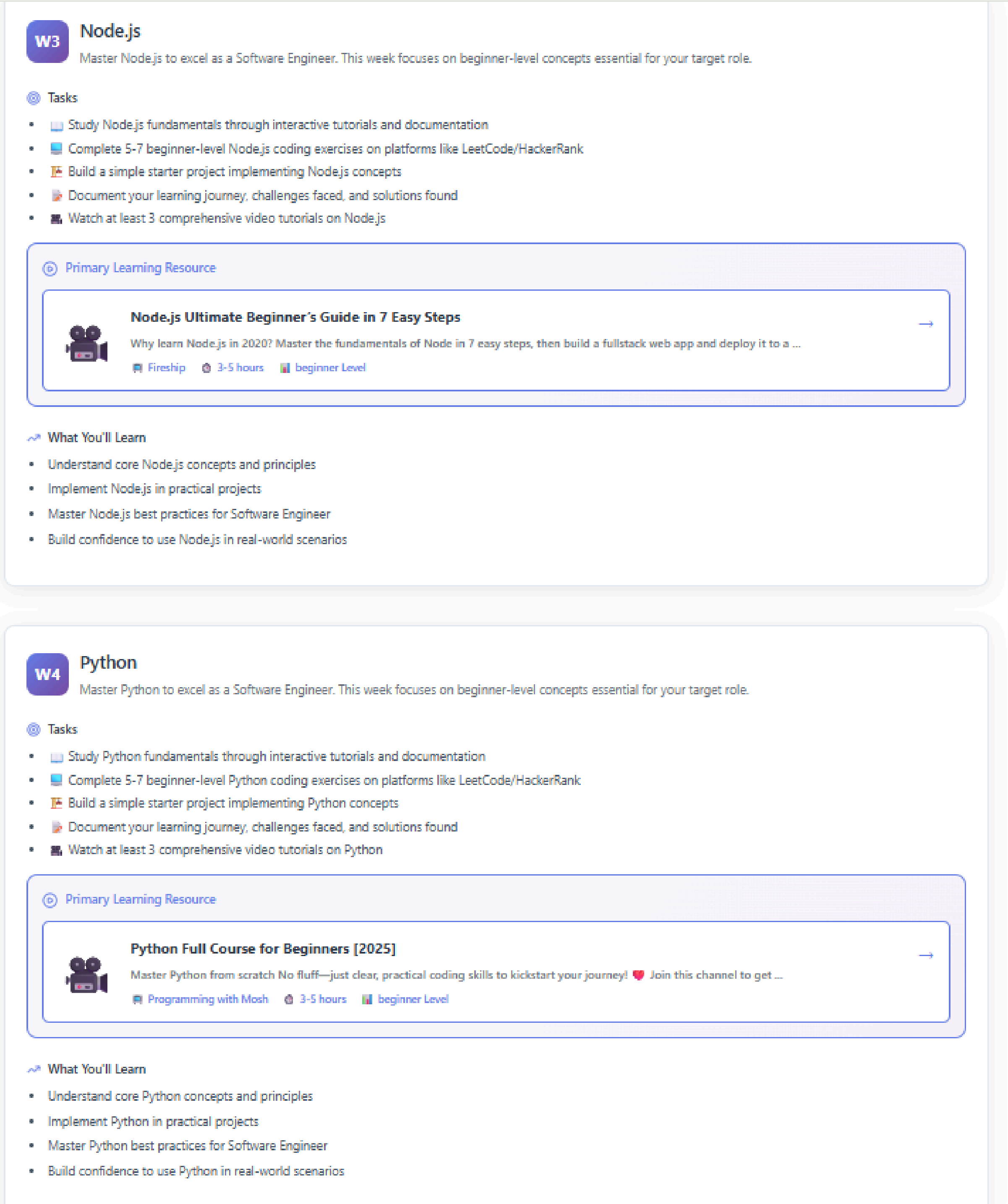
- Understand core JavaScript concepts and principles
- Implement JavaScript in practical projects
- Master JavaScript best practices for Software Engineer
- Build confidence to use JavaScript in real-world scenarios

W2 **React**

Master React to excel as a Software Engineer. This week focuses on beginner-level concepts essential for your target role.

Tasks

- 📖 Study React fundamentals through interactive tutorials and documentation
- 💻 Complete 5-7 beginner-level React coding exercises on platforms like LeetCode/HackerRank
- 🏗 Build a simple starter project implementing React concepts
- 📝 Document your learning journey, challenges faced, and solutions found
- 🎥 Watch at least 3 comprehensive video tutorials on React



W3

Node.js

Master Node.js to excel as a Software Engineer. This week focuses on beginner-level concepts essential for your target role.

Tasks

- 📖 Study Node.js fundamentals through interactive tutorials and documentation
- 💻 Complete 5-7 beginner-level Node.js coding exercises on platforms like LeetCode/HackerRank
- 🏗️ Build a simple starter project implementing Node.js concepts
- 📝 Document your learning journey, challenges faced, and solutions found
- 🎥 Watch at least 3 comprehensive video tutorials on Node.js

Primary Learning Resource

Node.js Ultimate Beginner's Guide in 7 Easy Steps

Why learn Node.js in 2020? Master the fundamentals of Node in 7 easy steps, then build a fullstack web app and deploy it to a ...

📺 Fireship ⏱ 3-5 hours 📊 beginner Level

→

What You'll Learn

- Understand core Node.js concepts and principles
- Implement Node.js in practical projects
- Master Node.js best practices for Software Engineer
- Build confidence to use Node.js in real-world scenarios

W4

Python

Master Python to excel as a Software Engineer. This week focuses on beginner-level concepts essential for your target role.

Tasks

- 📖 Study Python fundamentals through interactive tutorials and documentation
- 💻 Complete 5-7 beginner-level Python coding exercises on platforms like LeetCode/HackerRank
- 🏗️ Build a simple starter project implementing Python concepts
- 📝 Document your learning journey, challenges faced, and solutions found
- 🎥 Watch at least 3 comprehensive video tutorials on Python

Primary Learning Resource

Python Full Course for Beginners [2025]

Master Python from scratch No fluff—just clear, practical coding skills to kickstart your journey! ❤️ Join this channel to get ...

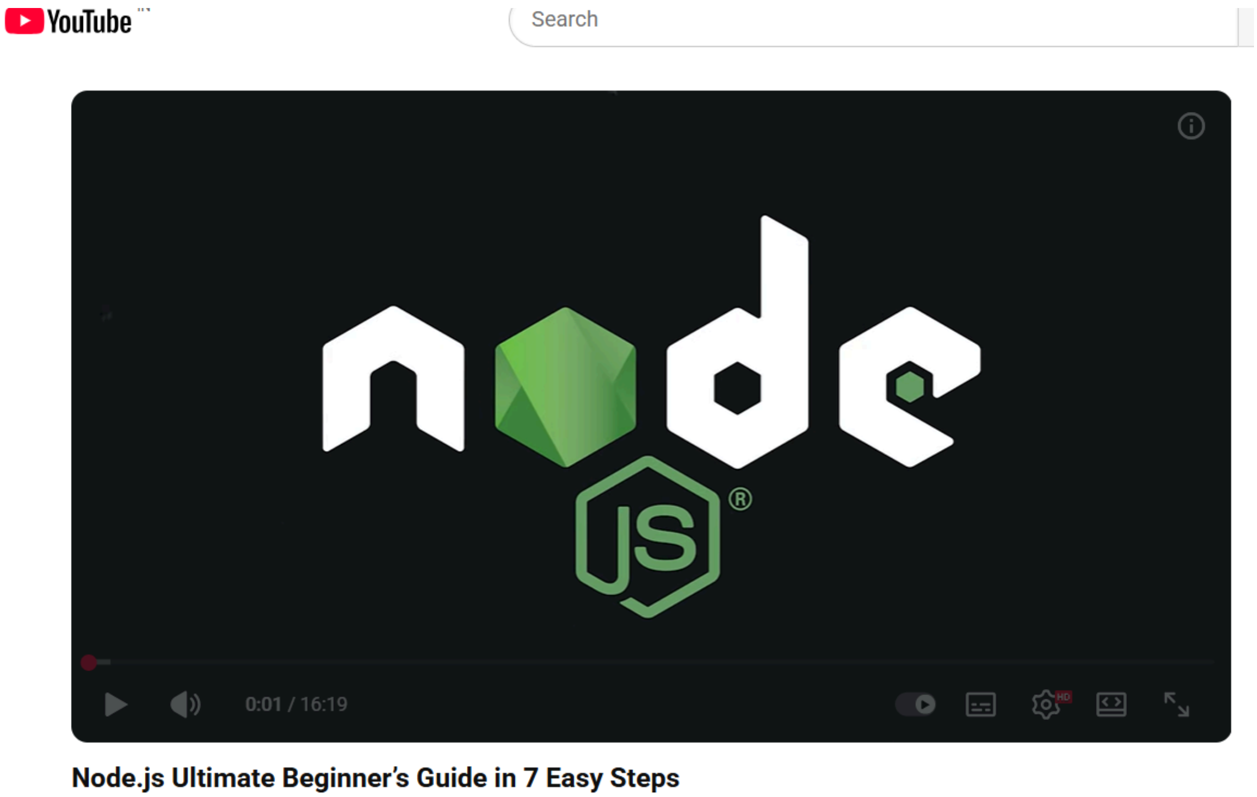
📺 Programming with Mosh ⏱ 3-5 hours 📊 beginner Level

→

What You'll Learn

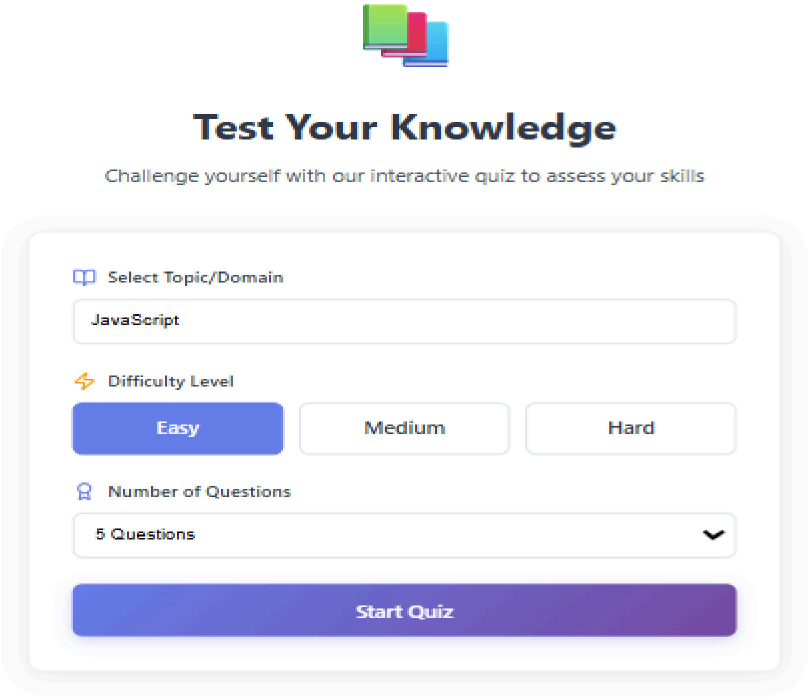
- Understand core Python concepts and principles
- Implement Python in practical projects
- Master Python best practices for Software Engineer
- Build confidence to use Python in real-world scenarios

Redirecting to Youtube:



Node.js Ultimate Beginner's Guide in 7 Easy Steps

Test your Knowledge page:



Test Your Knowledge

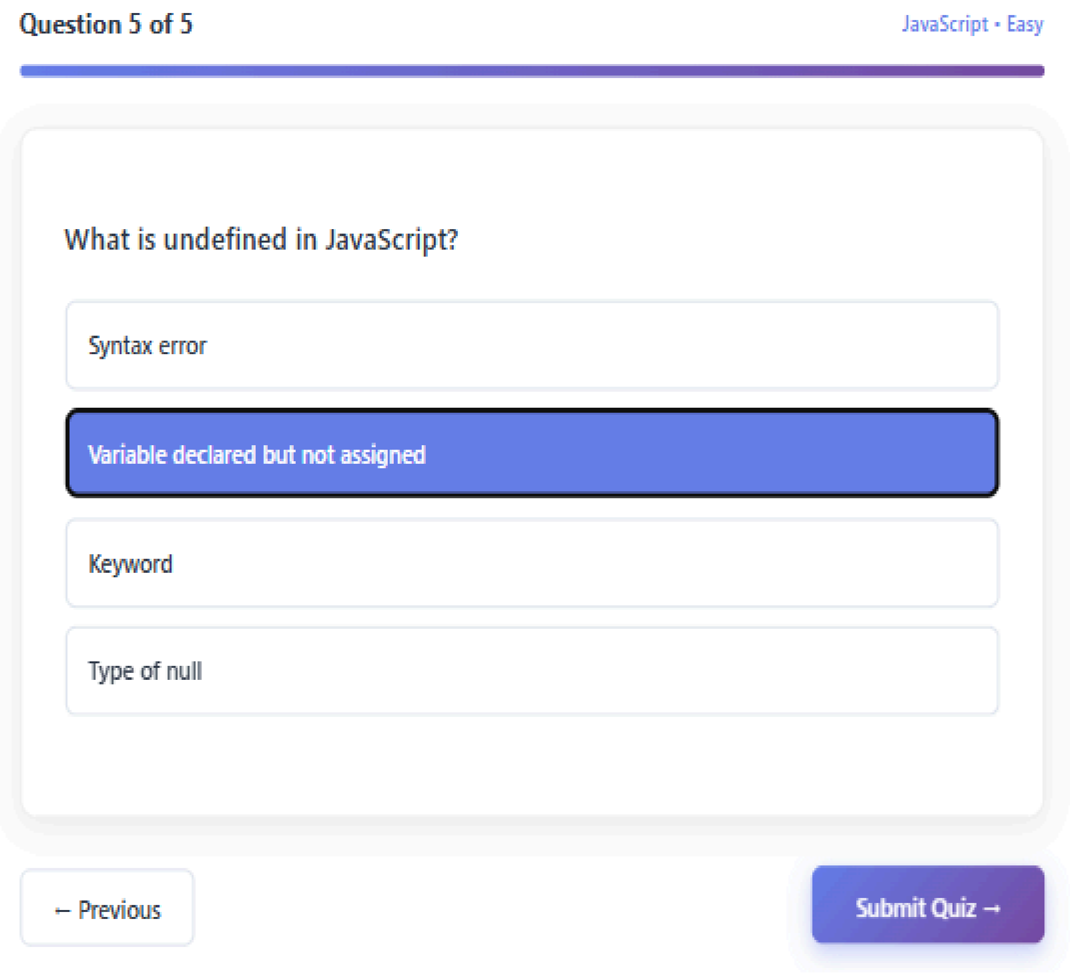
Challenge yourself with our interactive quiz to assess your skills

Select Topic/Domain: JavaScript

Difficulty Level: [✓ Easy] Medium Hard

Number of Questions: 5 Questions

Answering and Submitting Quiz and evaluation of quiz:



Question 5 of 5

JavaScript • Easy

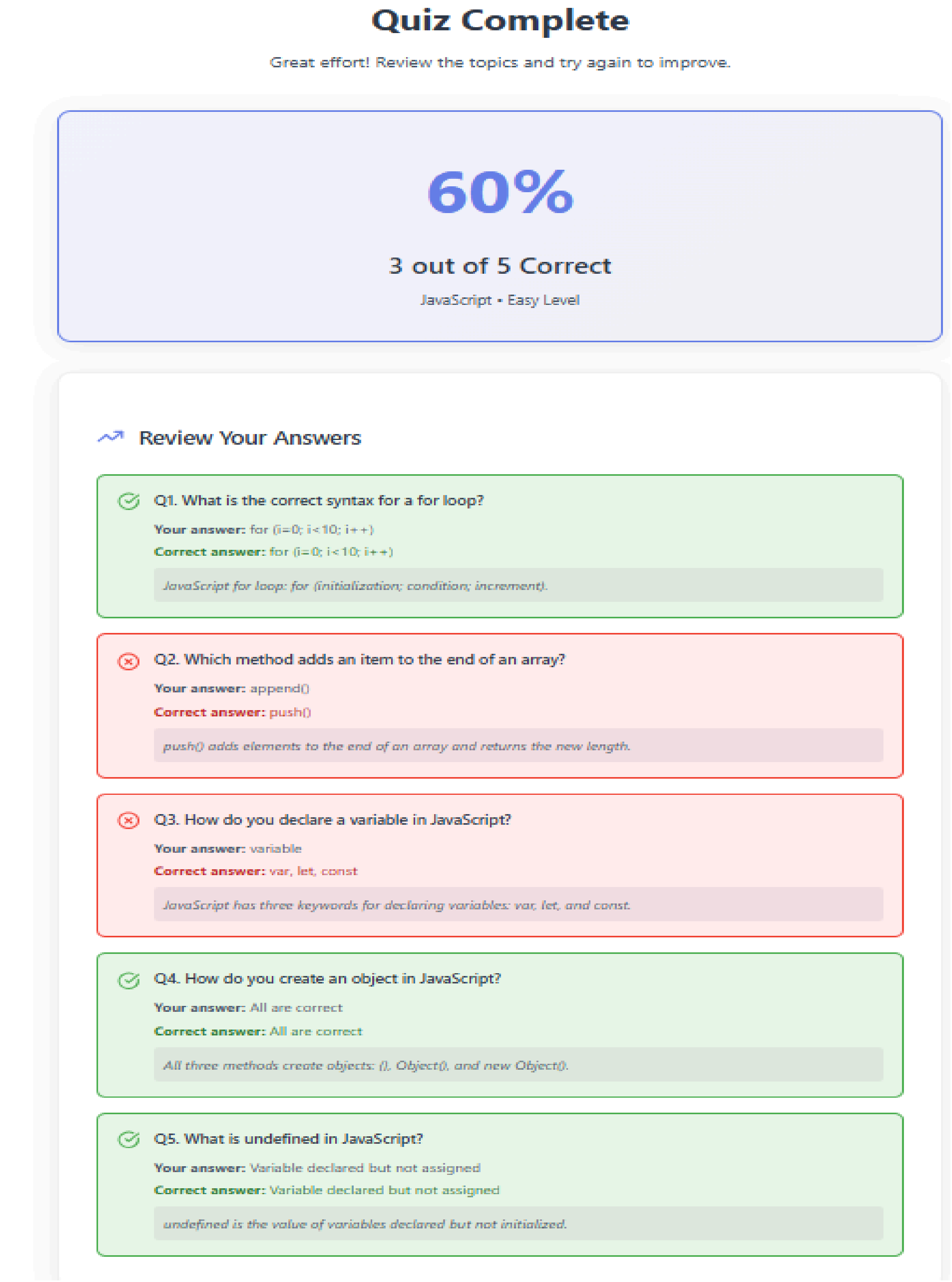
What is undefined in JavaScript?

Syntax error

Variable declared but not assigned

Keyword

Type of null



Quiz Complete

Great effort! Review the topics and try again to improve.

60%

3 out of 5 Correct

JavaScript • Easy Level

Review Your Answers

✓ Q1. What is the correct syntax for a for loop?

Your answer: for (i=0; i<10; i++)

Correct answer: for (i=0; i<10; i++)

JavaScript for loop: for (initialization; condition; increment).

✗ Q2. Which method adds an item to the end of an array?

Your answer: append()

Correct answer: push()

push() adds elements to the end of an array and returns the new length.

✗ Q3. How do you declare a variable in JavaScript?

Your answer: variable

Correct answer: var, let, const

JavaScript has three keywords for declaring variables: var, let, and const.

✓ Q4. How do you create an object in JavaScript?

Your answer: All are correct

Correct answer: All are correct

All three methods create objects: {}, Object(), and new Object().

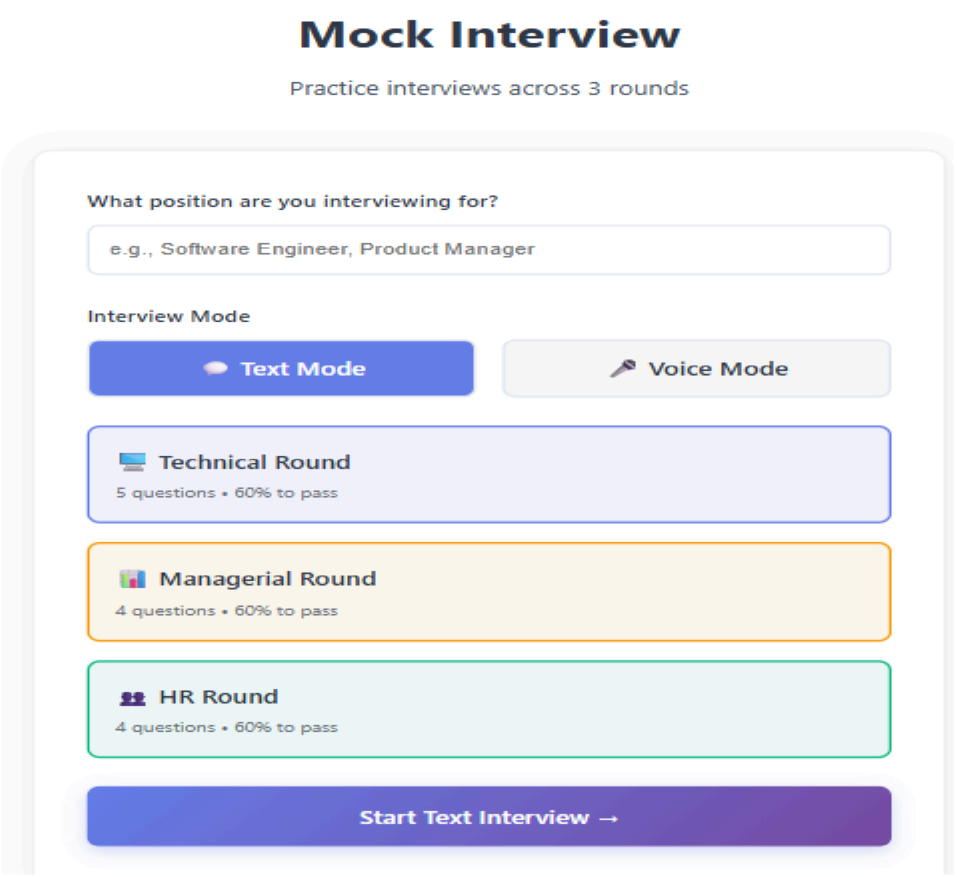
✓ Q5. What is undefined in JavaScript?

Your answer: Variable declared but not assigned

Correct answer: Variable declared but not assigned

undefined is the value of variables declared but not initialized.

Interview Section wrt Text and Voice mode:



Mock Interview

Practice interviews across 3 rounds

What position are you interviewing for?

e.g., Software Engineer, Product Manager

Interview Mode

💬 Text Mode | 🎤 Voice Mode

🖥 Technical Round
5 questions • 60% to pass

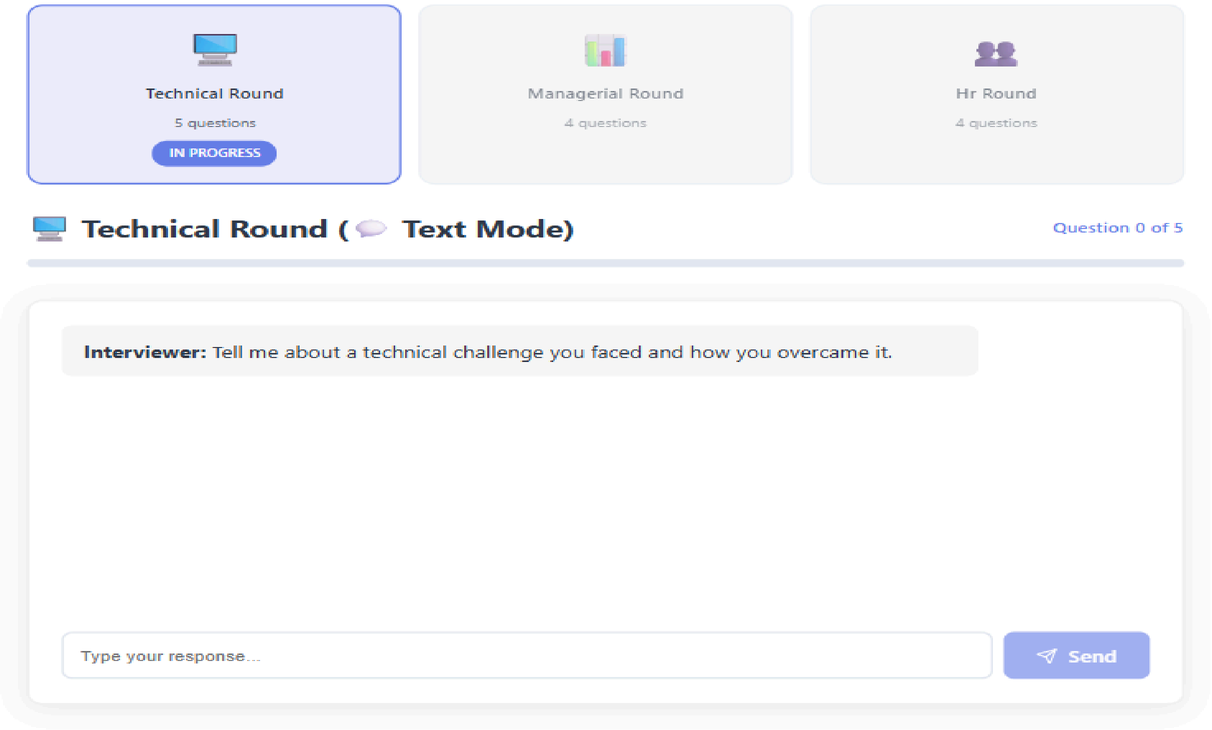
📊 Managerial Round
4 questions • 60% to pass

👥 HR Round
4 questions • 60% to pass

Start Text Interview →

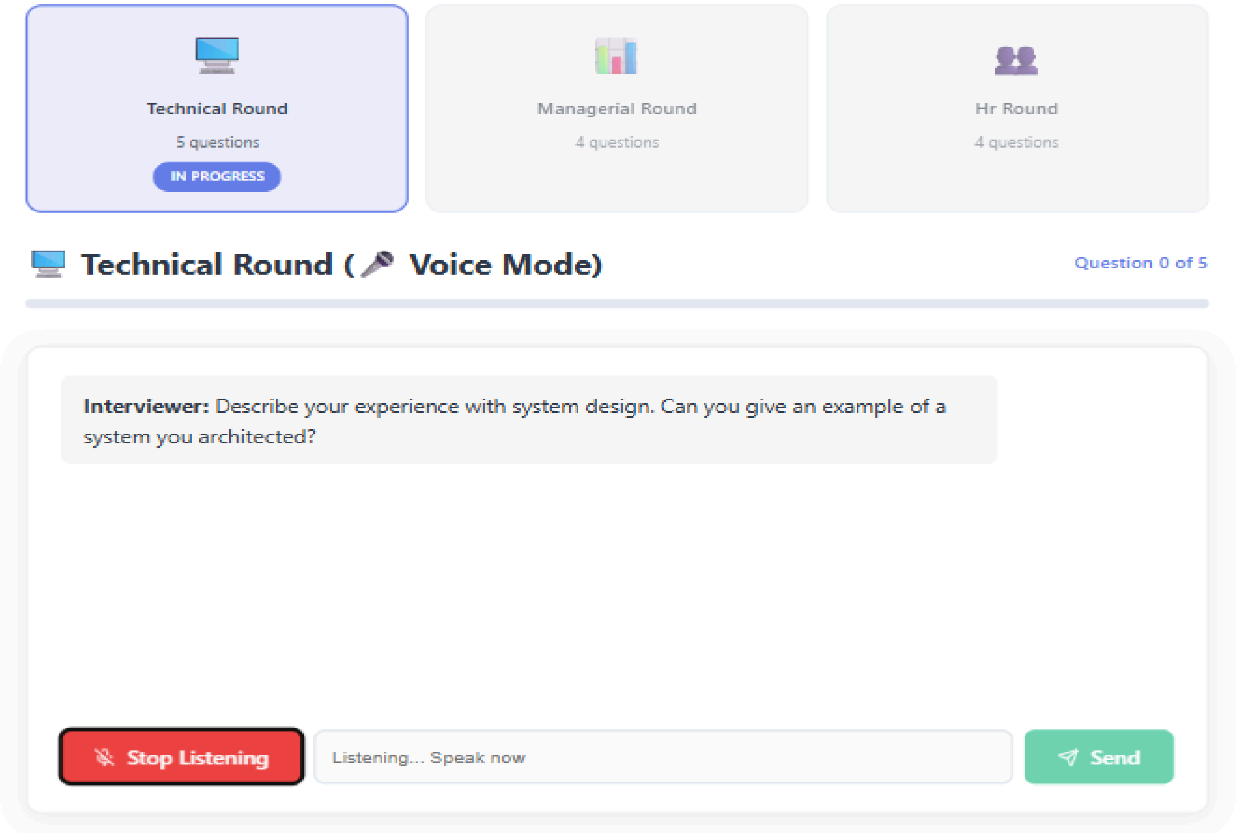
KEY POINTS:

- After achieving the qualifying score, the user receives a detailed overview of areas that need improvement.
- The system identifies skill gaps and provides clear, personalized recommendations.
- A structured learning path is generated to help the user strengthen weak areas and prepare more effectively for future interviews.



Technical Round (💬 Text Mode)

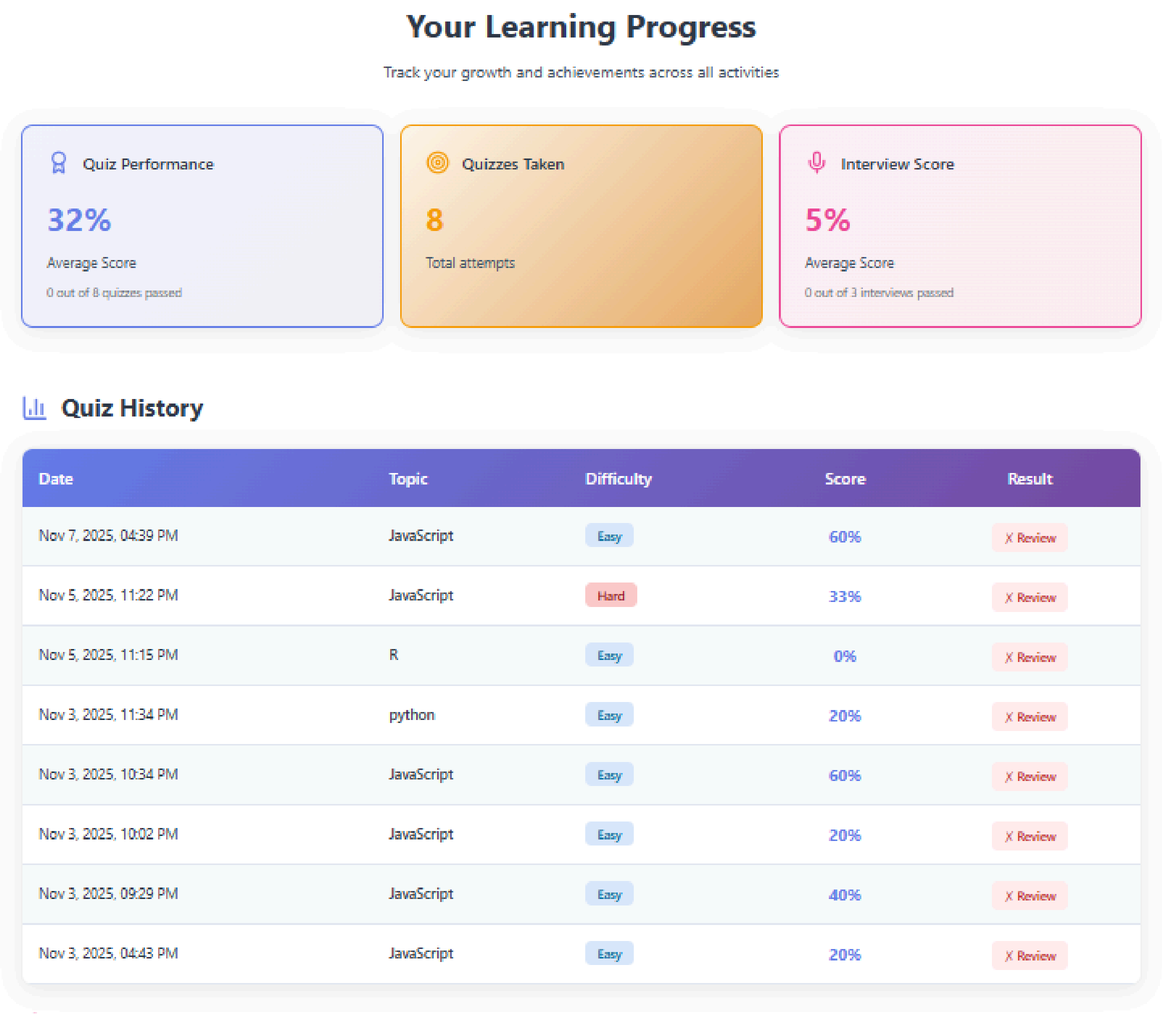
Interviewer: Tell me about a technical challenge you faced and how you overcame it.



Technical Round (🎤 Voice Mode)

Interviewer: Describe your experience with system design. Can you give an example of a system you architected?

Learning Progress Page:



Your Learning Progress

Track your growth and achievements across all activities

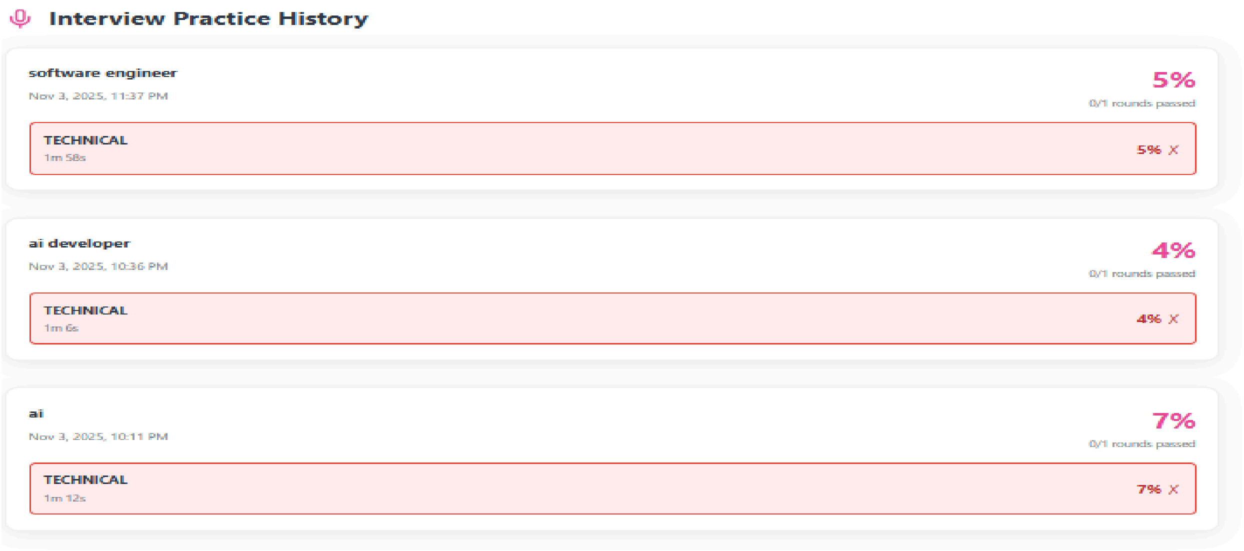
Quiz Performance
32%
Average Score
0 out of 8 quizzes passed

Quizzes Taken
8
Total attempts

Interview Score
5%
Average Score
0 out of 3 interviews passed

Quiz History

Date	Topic	Difficulty	Score	Result
Nov 7, 2025, 04:39 PM	JavaScript	Easy	60%	✗ Review
Nov 5, 2025, 11:22 PM	JavaScript	Hard	33%	✗ Review
Nov 5, 2025, 11:15 PM	R	Easy	0%	✗ Review
Nov 3, 2025, 11:34 PM	python	Easy	20%	✗ Review
Nov 3, 2025, 10:34 PM	JavaScript	Easy	60%	✗ Review
Nov 3, 2025, 10:02 PM	JavaScript	Easy	20%	✗ Review
Nov 3, 2025, 09:29 PM	JavaScript	Easy	40%	✗ Review
Nov 3, 2025, 04:43 PM	JavaScript	Easy	20%	✗ Review



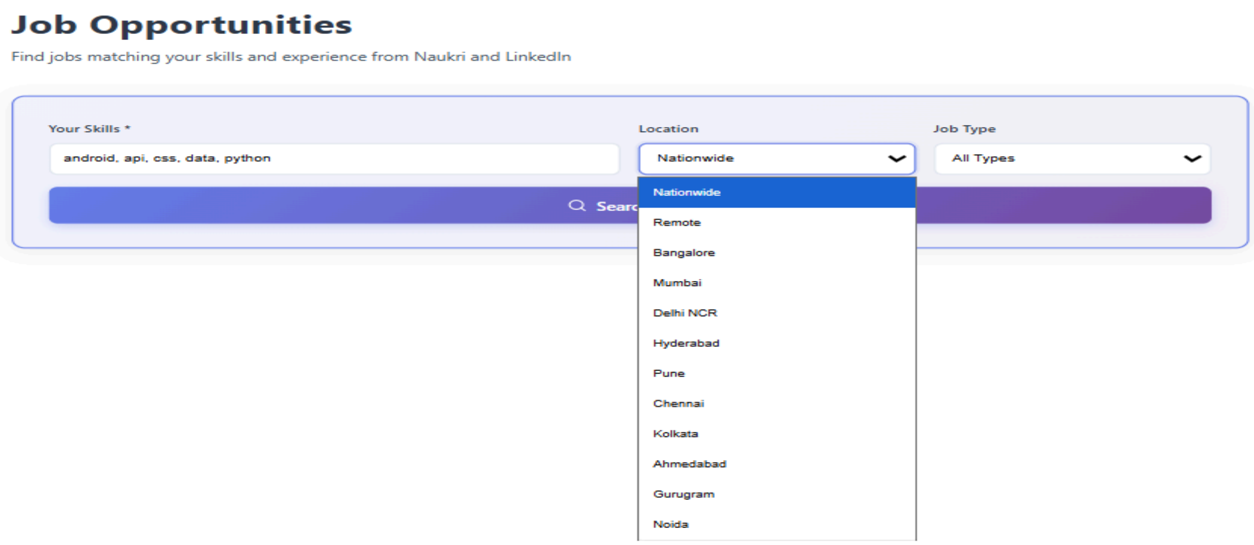
Interview Practice History

software engineer
Nov 3, 2025, 11:37 PM
5%
0/1 rounds passed
TECHNICAL
1m 58s
5% ✗

ai developer
Nov 3, 2025, 10:36 PM
4%
0/1 rounds passed
TECHNICAL
1m 6s
4% ✗

ai
Nov 3, 2025, 10:11 PM
7%
0/1 rounds passed
TECHNICAL
1m 12s
7% ✗

Job Selection Page and redirecting to Apply now page:



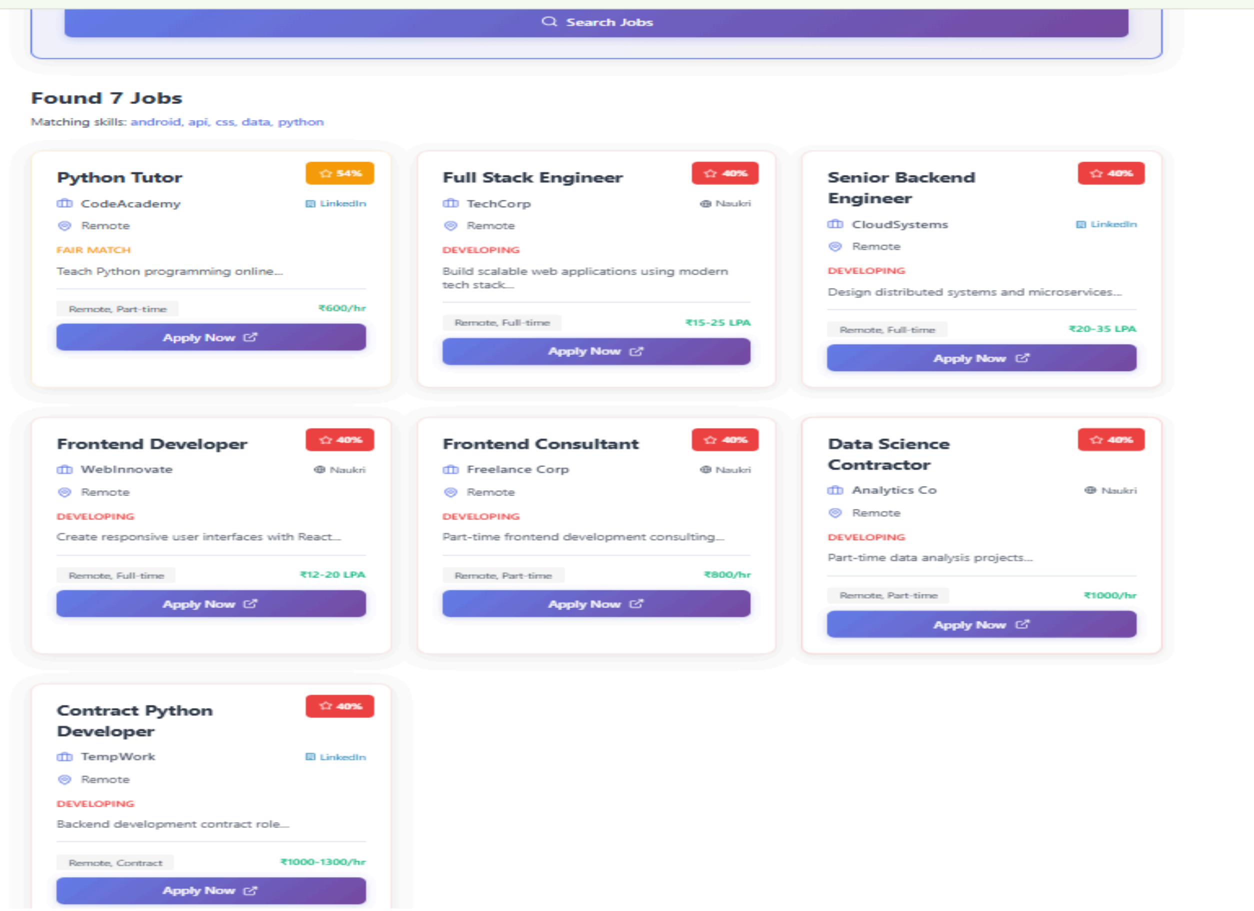
Job Opportunities

Find jobs matching your skills and experience from Naukri and LinkedIn

Your Skills *: android, api, css, data, python

Location: Nationwide

Job Type: All Types



Found 7 Jobs

Matching skills: android, api, css, data, python

Python Tutor — 54% — CodeAcademy — LinkedIn — Remote
FAIR MATCH
Teach Python programming online...
Remote, Part-time — ₹600/hr

Full Stack Engineer — 40% — TechCorp — Naukri — Remote
DEVELOPING
Build scalable web applications using modern tech stack...
Remote, Full-time — ₹15-25 LPA

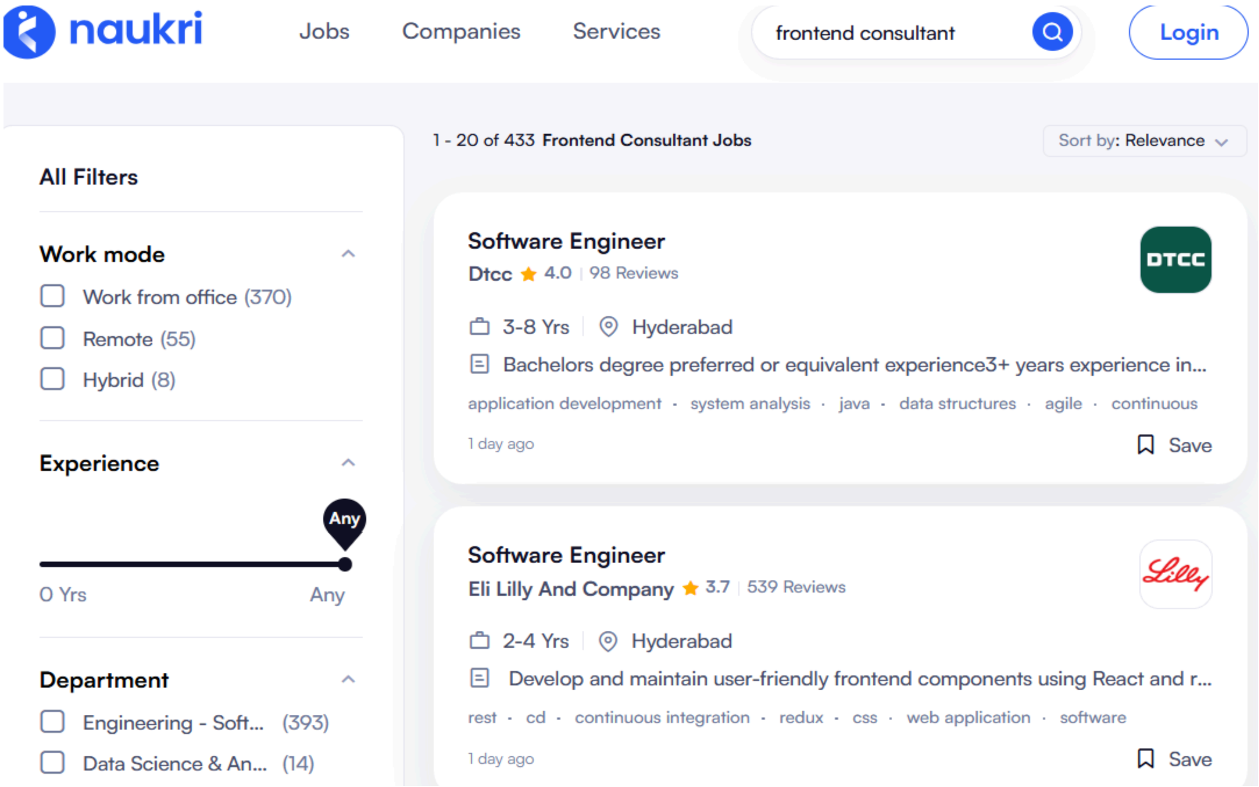
Senior Backend Engineer — 40% — CloudSystems — LinkedIn — Remote
DEVELOPING
Design distributed systems and microservices...
Remote, Full-time — ₹20-35 LPA

Frontend Developer — 40% — WebInnovate — Naukri — Remote
DEVELOPING
Create responsive user interfaces with React...
Remote, Full-time — ₹12-20 LPA

Frontend Consultant — 40% — Freelance Corp — Naukri — Remote
DEVELOPING
Part-time frontend development consulting...
Remote, Part-time — ₹800/hr

Data Science Contractor — 40% — Analytics Co — Naukri — Remote
DEVELOPING
Part-time data analysis projects...
Remote, Part-time — ₹1000/hr

Contract Python Developer — 40% — TempWork — LinkedIn — Remote
DEVELOPING
Backend development contract role...
Remote, Contract — ₹1000-1300/hr



frontend consultant

1 - 20 of 433 **Frontend Consultant Jobs**

Work mode

- □ Work from office (370)
- □ Remote (55)
- □ Hybrid (8)

Experience

Any

0 Yrs — Any

Department

- □ Engineering - Soft... (393)
- □ Data Science & An... (14)

Software Engineer

Dtcc ★ 4.0 | 98 Reviews

3-8 Yrs | Hyderabad

Bachelors degree preferred or equivalent experience3+ years experience in...

application development · system analysis · java · data structures · agile · continuous

1 day ago

Software Engineer

Eli Lilly And Company ★ 3.7 | 539 Reviews

2-4 Yrs | Hyderabad

Develop and maintain user-friendly frontend components using React and r...

rest · cd · continuous integration · redux · css · web application · software

1 day ago

Conclusion:

Vidyamitra is an AI-powered career guidance platform that helps students and professionals assess, enhance, and monitor their employability. It evaluates resumes, identifies skill gaps, and generates personalized training plans using AI and data analytics. With modules for resume parsing, mock interviews, and skill evaluation powered by GPT-4 and FastAPI, it offers adaptive and real-time guidance for effective career development.

The React-based frontend ensures an engaging user experience with features for tracking progress, visualizing skills, and accessing AI-driven insights. Integrated APIs like OpenAI, YouTube, Supabase, enable intelligent recommendations, secure data handling, and real-time updates. Vidyamitra thus bridges education and employability, transforming traditional mentoring into an interactive, automated, and data-driven experience.