



CareerPilot API Documentation

Version: 1.0.0

Base URL: `http://localhost:5000/api`

Authentication: JWT Bearer Token



Table of Contents

- [Authentication](#)
 - [Register](#)
 - [Login](#)
 - [Logout](#)
 - [Refresh Token](#)
 - [User Management](#)
 - [Get Profile](#)
 - [Update Profile](#)
 - [Delete Account](#)
 - [CV Management](#)
 - [Upload CV](#)
 - [Parse CV](#)
 - [Get CV Details](#)
 - [Delete CV](#)
 - [Career Recommendations](#)
 - [Get Recommendations](#)
 - [Get Career Details](#)
 - [Get Skill Gap Analysis](#)
 - [Roadmap Generation](#)
 - [Generate Roadmap](#)
 - [Get Roadmap](#)
 - [Update Roadmap Progress](#)
 - [AI Chat](#)
 - [Send Message](#)
 - [Get Chat History](#)
 - [Clear Chat](#)
 - [Progress Tracking](#)
 - [Get Progress](#)
 - [Update Progress](#)
 - [Get Statistics](#)
 - [Skills](#)
 - [Get All Skills](#)
 - [Add Skill](#)
 - [Remove Skill](#)
-



Authentication

All protected endpoints require JWT token in Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

1. Register User

Endpoint: POST /api/auth/register

Description: Create a new user account

Headers:

```
{  
  "Content-Type": "application/json"  
}
```

Request Body:

```
{  
  "email": "john.doe@example.com",  
  "password": "SecurePassword123!",  
  "firstName": "John",  
  "lastName": "Doe",  
  "phone": "+1234567890"  
}
```

Success Response (201):

```
{  
  "success": true,  
  "message": "User registered successfully",  
  "data": {  
    "user": {  
      "id": "uuid-here",  
      "email": "john.doe@example.com",  
      "firstName": "John",  
      "lastName": "Doe",  
      "createdAt": "2025-11-18T10:30:00Z"  
    },  
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "expiresIn": "7d"  
  }  
}
```

Error Response (400):

```
{  
  "success": false,  
  "error": "Email already exists"  
}
```

2. Login User

Endpoint: POST /api/auth/login

Description: Authenticate user and receive JWT token

Request Body:

```
{  
  "email": "john.doe@example.com",  
  "password": "SecurePassword123!"  
}
```

Success Response (200):

```
{  
  "success": true,  
  "message": "Login successful",  
  "data": {  
    "user": {  
      "id": "uuid-here",  
      "email": "john.doe@example.com",  
      "firstName": "John",  
      "lastName": "Doe"  
    },  
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "expiresIn": "7d"  
  }  
}
```

Error Response (401):

```
{  
  "success": false,  
  "error": "Invalid credentials"  
}
```

3. Logout User

Endpoint: POST /api/auth/logout

Description: Logout user and invalidate token

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{  
  "success": true,
```

```
        "message": "Logged out successfully"
    }
```

4. Refresh Token

Endpoint: POST /api/auth/refresh

Description: Refresh JWT token before expiration

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "token": "new_jwt_token_here",
    "expiresIn": "7d"
  }
}
```

User Management

5. Get User Profile

Endpoint: GET /api/user/profile

Description: Get complete user profile with skills and preferences

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "user": {
      "id": "uuid-here",
      "email": "john.doe@example.com",
      "firstName": "John",
      "lastName": "Doe",
      "phone": "+1234567890",
      "profile": {
        "bio": "Software developer passionate about AI",
        "education": "Bachelor's in Computer Science",
        "experience": "2 years",
        "currentRole": "Junior Developer",
        "skills": [
          "Python",
          "Machine Learning",
          "Data Structures"
        ]
      }
    }
  }
}
```

```
        "careerGoals": "Become a Senior Full-Stack Developer",
        "interests": ["Web Development", "AI/ML", "Cloud Computing"],
        "linkedinUrl": "https://linkedin.com/in/johndoe",
        "githubUrl": "https://github.com/johndoe"
    },
    "skills": [
        {
            "id": "skill-1",
            "name": "JavaScript",
            "level": "intermediate",
            "verified": true
        },
        {
            "id": "skill-2",
            "name": "React",
            "level": "intermediate",
            "verified": true
        }
    ],
    "cvUploaded": true,
    "hasRoadmap": true,
    "createdAt": "2025-11-18T10:30:00Z"
}
}
```

6. Update User Profile

Endpoint: PUT /api/user/profile

Description: Update user profile information

Headers:

```
Authorization: Bearer <token>
Content-Type: application/json
```

Request Body:

```
{
    "firstName": "John",
    "lastName": "Smith",
    "phone": "+1234567890",
    "profile": {
        "bio": "Updated bio",
        "education": "Master's in CS",
        "experience": "3 years",
        "currentRole": "Mid-Level Developer",
        "careerGoals": "Become AI Engineer",
        "interests": ["AI", "ML", "Data Science"],
        "linkedinUrl": "https://linkedin.com/in/johnsmith",
        "githubUrl": "https://github.com/johnsmith"
    }
}
```

```
    }
}
```

Success Response (200):

```
{
  "success": true,
  "message": "Profile updated successfully",
  "data": {
    "user": { /* updated user object */ }
  }
}
```

7. Delete User Account

Endpoint: DELETE /api/user/account

Description: Permanently delete user account and all associated data

Headers:

```
Authorization: Bearer <token>
```

Request Body:

```
{
  "password": "SecurePassword123!",
  "confirmation": "DELETE"
}
```

Success Response (200):

```
{
  "success": true,
  "message": "Account deleted successfully"
}
```

CV Management

8. Upload CV

Endpoint: POST /api/cv/upload

Description: Upload CV file (PDF/DOCX, max 5MB)

Headers:

```
Authorization: Bearer <token>
Content-Type: multipart/form-data
```

Request Body (Form Data):

file: <CV_FILE>

Success Response (201):

```
{  
  "success": true,  
  "message": "CV uploaded successfully",  
  "data": {  
    "cv": {  
      "id": "cv-uuid",  
      "fileName": "john_doe_cv.pdf",  
      "fileSize": 245678,  
      "uploadedAt": "2025-11-18T11:00:00Z",  
      "status": "uploaded"  
    }  
  }  
}
```

Error Response (400):

```
{  
  "success": false,  
  "error": "File size exceeds 5MB limit"  
}
```

9. Parse CV**Endpoint:** POST /api/cv/parse**Description:** Parse uploaded CV and extract skills, experience, and education**Headers:**

Authorization: Bearer <token>

Request Body:

```
{  
  "cvId": "cv-uuid"  
}
```

Success Response (200):

```
{  
  "success": true,  
  "message": "CV parsed successfully",  
  "data": {  
    "parsed": {  
    }  
  }  
}
```

```

"personalInfo": {
    "name": "John Doe",
    "email": "john.doe@example.com",
    "phone": "+1234567890",
    "location": "New York, USA"
},
"skills": [
    "JavaScript", "React", "Node.js", "PostgreSQL",
    "Git", "Docker", "AWS"
],
"experience": [
    {
        "title": "Junior Developer",
        "company": "Tech Corp",
        "duration": "2 years",
        "description": "Worked on full-stack web applications"
    }
],
"education": [
    {
        "degree": "Bachelor of Science in Computer Science",
        "institution": "State University",
        "year": "2023"
    }
],
"certifications": [
    "AWS Cloud Practitioner"
],
"languages": ["English", "Spanish"]
},
"aiAnalysis": {
    "strengthAreas": ["Frontend Development", "JavaScript Ecosystem"],
    "weakAreas": ["System Design", "Database Optimization"],
    "experienceLevel": "Junior",
    "recommendedPaths": ["Full-Stack Developer", "Frontend Specialist"]
}
}
}

```

10. Get CV Details

Endpoint: GET /api/cv/:cvId

Description: Get CV metadata and parsed information

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{  
  "success": true,  
  "data": {  
    "cv": {  
      "id": "cv-uuid",  
      "fileName": "john_doe_cv.pdf",  
      "fileSize": 245678,  
      "uploadedAt": "2025-11-18T11:00:00Z",  
      "parsedAt": "2025-11-18T11:05:00Z",  
      "status": "parsed",  
      "downloadUrl": "/api/cv/download/cv-uuid"  
    }  
  }  
}
```

11. Delete CV

Endpoint: DELETE /api/cv/:cvId

Description: Delete uploaded CV

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{  
  "success": true,  
  "message": "CV deleted successfully"  
}
```

🎯 Career Recommendations

12. Get Career Recommendations

Endpoint: POST /api/career/recommend

Description: Get AI-powered career path recommendations based on user profile and CV

Headers:

```
Authorization: Bearer <token>  
Content-Type: application/json
```

Request Body:

```
{  
  "includeSkillGap": true,  
  "includeSalaryData": true,
```

```
        "includeMarketDemand": true
    }
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "recommendations": [
      {
        "id": "career-1",
        "title": "Full-Stack Developer",
        "matchScore": 85,
        "description": "Develop both frontend and backend applications",
        "reasoning": "Your JavaScript and React skills align well with this path",
        "requiredSkills": [
          "JavaScript", "React", "Node.js", "Databases",
          "REST APIs", "Git", "Testing"
        ],
        "yourSkills": [
          "JavaScript", "React", "Git"
        ],
        "missingSkills": [
          "Node.js (Backend)", "Database Design", "API Development", "Testing Frameworks"
        ],
        "recommendedCertifications": [
          "AWS Certified Developer",
          "MongoDB Developer Certification"
        ],
        "averageSalary": {
          "min": 70000,
          "max": 120000,
          "currency": "USD"
        },
        "marketDemand": "High",
        "growthPotential": "Excellent"
      },
      {
        "id": "career-2",
        "title": "Frontend Developer",
        "matchScore": 92,
        "description": "Build user interfaces and user experiences",
        "reasoning": "Strong React skills make you an ideal candidate",
        "requiredSkills": [
          "HTML", "CSS", "JavaScript", "React", "TypeScript",
          "Responsive Design", "Testing"
        ],
        "yourSkills": [
          "HTML", "CSS", "JavaScript", "React"
        ],
        "missingSkills": [
          "HTML", "CSS", "JavaScript", "React"
        ]
      }
    ]
  }
}
```

```

        "TypeScript", "Advanced CSS (Grid/Flexbox)", "Unit Testing"
    ],
    "recommendedCertifications": [
        "Meta Frontend Developer Professional Certificate"
    ],
    "averageSalary": {
        "min": 65000,
        "max": 110000,
        "currency": "USD"
    },
    "marketDemand": "Very High",
    "growthPotential": "Good"
},
{
    "id": "career-3",
    "title": "DevOps Engineer",
    "matchScore": 65,
    "description": "Automate deployment and infrastructure",
    "reasoning": "You have foundational skills, but need more infrastructure knowledge",
    "requiredSkills": [
        "Linux", "Docker", "Kubernetes", "CI/CD",
        "AWS/Azure", "Scripting", "Monitoring"
    ],
    "yourSkills": [
        "Git", "Basic AWS"
    ],
    "missingSkills": [
        "Docker", "Kubernetes", "CI/CD Pipelines",
        "Infrastructure as Code", "Linux Administration"
    ],
    "recommendedCertifications": [
        "AWS Solutions Architect",
        "Kubernetes Administrator (CKA)"
    ],
    "averageSalary": {
        "min": 80000,
        "max": 140000,
        "currency": "USD"
    },
    "marketDemand": "High",
    "growthPotential": "Excellent"
}
],
{
    "topRecommendation": "Frontend Developer",
    "analysisDate": "2025-11-18T12:00:00Z"
}
}

```

13. Get Career Path Details

Endpoint: GET /api/career/:careeId

Description: Get detailed information about a specific career path

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "career": {
      "id": "career-1",
      "title": "Full-Stack Developer",
      "description": "Design and develop both client and server software",
      "responsibilities": [
        "Build responsive web applications",
        "Develop RESTful APIs",
        "Database design and optimization",
        "Write clean, maintainable code",
        "Collaborate with design and product teams"
      ],
      "skillsRequired": {
        "technical": [
          "JavaScript/TypeScript",
          "React/Vue/Angular",
          "Node.js/Python/Java",
          "SQL/NoSQL databases",
          "Git version control",
          "REST API design",
          "Testing frameworks"
        ],
        "soft": [
          "Problem-solving",
          "Communication",
          "Time management",
          "Teamwork"
        ]
      },
      "educationPath": [
        "Bachelor's in Computer Science (or equivalent)",
        "Online courses (freeCodeCamp, Udemy, Coursera)",
        "Bootcamps (optional)"
      ],
      "certifications": [
        "AWS Certified Developer",
        "MongoDB Developer",
        "Google Cloud Professional"
      ],
      "careerProgression": [
        {
          "level": "Junior",
          "experience": "0-2 years"
        },
        {
          "level": "Mid-level",
          "experience": "2-5 years"
        },
        {
          "level": "Senior",
          "experience": "5+ years"
        }
      ]
    }
  }
}
```

```

        "duration": "0-2 years",
        "salary": "$50k-$70k"
    },
    {
        "level": "Mid-Level",
        "duration": "2-5 years",
        "salary": "$70k-$100k"
    },
    {
        "level": "Senior",
        "duration": "5+ years",
        "salary": "$100k-$150k"
    }
],
"marketInsights": {
    "demand": "Very High",
    "growth": "+15% annually",
    "topEmployers": ["Google", "Amazon", "Microsoft", "Startups"]
}
}
}
}

```

14. Get Skill Gap Analysis

Endpoint: GET /api/career/skill-gap/:careerId

Description: Detailed skill gap analysis for a specific career path

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "analysis": {
      "careerId": "career-1",
      "careerTitle": "Full-Stack Developer",
      "overallMatch": 85,
      "skillBreakdown": {
        "mastered": [
          { "skill": "JavaScript", "yourLevel": "intermediate" },
          { "skill": "React", "yourLevel": "intermediate" }
        ],
        "learning": [
          { "skill": "Node.js", "yourLevel": "beginner", "targetLevel": "intermediate" }
        ],
        "missing": [
          { "skill": "PostgreSQL", "priority": "high" },

```

```

        {
          "skill": "Docker", "priority": "medium" },
          {
            "skill": "Testing", "priority": "high" }
        ],
      },
      "timeToReady": "3-6 months",
      "recommendedActions": [
        {
          "action": "Complete Node.js backend course",
          "priority": "high",
          "estimatedTime": "4 weeks"
        },
        {
          "action": "Build 2-3 full-stack projects",
          "priority": "high",
          "estimatedTime": "6 weeks"
        },
        {
          "action": "Learn PostgreSQL fundamentals",
          "priority": "high",
          "estimatedTime": "2 weeks"
        }
      ]
    }
  }
}

```

Roadmap Generation

15. Generate Learning Roadmap

Endpoint: POST /api/roadmap/generate

Description: Generate personalized learning roadmap for selected career path

Headers:

```

Authorization: Bearer <token>
Content-Type: application/json

```

Request Body:

```

{
  "careerId": "career-1",
  "duration": "6 months",
  "intensity": "moderate",
  "focusAreas": ["backend", "databases", "testing"]
}

```

Success Response (201):

```
{
  "success": true,
  "message": "Roadmap generated successfully",
  "data": {
    "roadmap": {
      "id": "roadmap-uuid",
      "careerId": "career-1",
      "careerTitle": "Full-Stack Developer",
      "duration": "6 months",
      "startDate": "2025-11-18",
      "targetEndDate": "2026-05-18",
      "phases": [
        {
          "phase": 1,
          "title": "Foundation Month",
          "duration": "4 weeks",
          "weeks": [
            {
              "week": 1,
              "focus": "Node.js Basics",
              "tasks": [
                {
                  "id": "task-1",
                  "title": "Complete Node.js Introduction Course",
                  "type": "course",
                  "link": "https://nodejs.org/en/learn",
                  "estimatedHours": 10,
                  "completed": false
                },
                {
                  "id": "task-2",
                  "title": "Build a simple Express server",
                  "type": "project",
                  "description": "Create REST API with basic routes",
                  "estimatedHours": 5,
                  "completed": false
                }
              ],
              "learningGoals": [
                "Understand Node.js event loop",
                "Master Express routing",
                "Handle async operations"
              ]
            },
            {
              "week": 2,
              "focus": "Database Fundamentals",
              "tasks": [
                {
                  "id": "task-3",
                  "title": "PostgreSQL Tutorial",
                  "type": "course",
                  "link": "https://www.postgresqltutorial.com",
                  "estimatedHours": 8,
                  "completed": false
                }
              ],
              "learningGoals": [
                "Understand PostgreSQL query language"
              ]
            }
          ],
          "estimatedDuration": "4 weeks"
        }
      ],
      "estimatedDuration": "6 months"
    }
  }
}
```

```
        "type": "course",
        "link": "https://www.postgresql.org/docs/tutorial/",
        "estimatedHours": 8,
        "completed": false
    },
    {
        "id": "task-4",
        "title": "Design database schema for blog app",
        "type": "project",
        "estimatedHours": 4,
        "completed": false
    }
],
"learningGoals": [
    "SQL query fundamentals",
    "Database design patterns",
    "Relationships and joins"
]
}
],
},
{
    "phase": 2,
    "title": "Integration Month",
    "duration": "4 weeks",
    "weeks": [ /* similar structure */ ]
},
{
    "phase": 3,
    "title": "Advanced Topics",
    "duration": "4 weeks",
    "weeks": [ /* similar structure */ ]
}
],
"milestones": [
    {
        "title": "Build first full-stack app",
        "targetDate": "2026-01-15",
        "completed": false
    },
    {
        "title": "Complete authentication system",
        "targetDate": "2026-02-15",
        "completed": false
    }
],
"resources": [
    {
        "title": "Node.js Documentation",
        "type": "documentation",
        "url": "https://nodejs.org/docs"
    },

```

```
{
  {
    "title": "The Complete Node.js Course",
    "type": "course",
    "url": "https://udemy.com/nodejs-course",
    "platform": "Udemy"
  }
],
"createdAt": "2025-11-18T12:30:00Z"
}
}
```

16. Get User Roadmap

Endpoint: GET /api/roadmap

Description: Get user's current learning roadmap with progress

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "roadmap": { /* roadmap object */ },
    "progress": {
      "overallCompletion": 35,
      "currentPhase": 2,
      "currentWeek": 6,
      "tasksCompleted": 14,
      "totalTasks": 40,
      "hoursSpent": 85,
      "estimatedHoursRemaining": 155
    }
  }
}
```

17. Update Roadmap Progress

Endpoint: PUT /api/roadmap/progress

Description: Mark tasks as complete and update progress

Headers:

```
Authorization: Bearer <token>
Content-Type: application/json
```

Request Body:

```
{
  "taskId": "task-1",
  "completed": true,
  "notes": "Completed the course, built a simple server",
  "hoursSpent": 12
}
```

Success Response (200):

```
{
  "success": true,
  "message": "Progress updated successfully",
  "data": {
    "task": {
      "id": "task-1",
      "completed": true,
      "completedAt": "2025-11-18T14:00:00Z"
    },
    "newProgress": {
      "overallCompletion": 38,
      "tasksCompleted": 15
    }
  }
}
```

💬 AI Chat

18. Send Chat Message

Endpoint: POST /api/chat/message

Description: Send message to AI career coach and get response

Headers:

```
Authorization: Bearer <token>
Content-Type: application/json
```

Request Body:

```
{
  "message": "What should I focus on this week?",
  "context": "roadmap"
}
```

Success Response (200):

```
{
  "success": true,
  "data": {
```

```
"message": {
    "id": "msg-uuid",
    "userMessage": "What should I focus on this week?",
    "aiResponse": "Based on your roadmap, you should focus on completing the Node.js basics course and building your first Express server. This week's goals include:\n\n1. Understanding the Node.js event loop\n2. Creating basic routes with Express\n3. Handling async operations\n\nI recommend dedicating 10-12 hours this week. Would you like specific resources?", 
    "timestamp": "2025-11-18T15:00:00Z",
    "contextUsed": ["roadmap", "current_progress", "user_skills"]
}
}
```

19. Get Chat History

Endpoint: GET /api/chat/history

Description: Retrieve chat conversation history

Headers:

```
Authorization: Bearer <token>
```

Query Parameters:

```
?limit=20&offset=0
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "messages": [
      {
        "id": "msg-1",
        "userMessage": "How do I start learning backend?",
        "aiResponse": "Great question! Here's a structured approach...",
        "timestamp": "2025-11-17T10:00:00Z"
      },
      {
        "id": "msg-2",
        "userMessage": "What should I focus on this week?",
        "aiResponse": "Based on your roadmap...",
        "timestamp": "2025-11-18T15:00:00Z"
      }
    ],
    "pagination": {
      "total": 45,
      "limit": 20,
      "offset": 0,
      "hasMore": true
    }
  }
}
```

```
        }
    }
}
```

20. Clear Chat History

Endpoint: DELETE /api/chat/history

Description: Clear all chat messages

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "message": "Chat history cleared successfully"
}
```

Progress Tracking

21. Get User Progress

Endpoint: GET /api/progress

Description: Get overall learning progress and statistics

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "progress": {
      "userId": "uuid-here",
      "roadmapProgress": 35,
      "skillsAcquired": 8,
      "coursesCompleted": 3,
      "projectsBuilt": 2,
      "certificationsEarned": 1,
      "weeklyStreak": 4,
      "totalHoursLearned": 85,
      "lastActivity": "2025-11-18T14:00:00Z",
      "skillBreakdown": [
        {
          "skill": "Node.js",

```

```

        "progress": 60,
        "level": "beginner"
    },
    {
        "skill": "PostgreSQL",
        "progress": 40,
        "level": "beginner"
    }
],
"weeklyActivity": [
    { "week": "2025-W45", "hoursSpent": 12 },
    { "week": "2025-W46", "hoursSpent": 15 },
    { "week": "2025-W47", "hoursSpent": 18 }
]
}
}
}
}

```

22. Update Progress

Endpoint: POST /api/progress/update

Description: Log learning activity and update progress

Headers:

```

Authorization: Bearer <token>
Content-Type: application/json

```

Request Body:

```

{
    "activityType": "course_completed",
    "details": {
        "courseName": "Node.js Fundamentals",
        "platform": "Udemy",
        "hoursSpent": 12,
        "skillsLearned": ["Node.js", "Express", "NPM"]
    }
}

```

Success Response (200):

```

{
    "success": true,
    "message": "Progress updated successfully",
    "data": {
        "newProgress": 38,
        "achievementUnlocked": "First Course Completed!"
    }
}

```

```
    }
}
```

23. Get User Statistics

Endpoint: GET /api/progress/statistics

Description: Get detailed learning analytics

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "data": {
    "statistics": {
      "overall": {
        "totalHours": 85,
        "tasksCompleted": 15,
        "currentStreak": 4,
        "longestStreak": 12
      },
      "thisWeek": {
        "hoursLearned": 18,
        "tasksCompleted": 3,
        "coursesFinished": 1
      },
      "thisMonth": {
        "hoursLearned": 65,
        "tasksCompleted": 12,
        "skillsGained": 3
      },
      "predictions": {
        "roadmapCompletionDate": "2026-04-20",
        "nextMilestone": "2026-01-15",
        "recommendedPace": "15 hours/week"
      }
    }
  }
}
```



24. Get All Skills

Endpoint: GET /api/skills

Description: Get list of all available skills in the system

Success Response (200):

```
{  
  "success": true,  
  "data": {  
    "skills": [  
      {  
        "id": "skill-1",  
        "name": "JavaScript",  
        "category": "Programming Language",  
        "popularity": 95  
      },  
      {  
        "id": "skill-2",  
        "name": "React",  
        "category": "Frontend Framework",  
        "popularity": 88  
      }  
    ]  
  }  
}
```

25. Add User Skill

Endpoint: POST /api/skills/add

Description: Add skill to user profile

Headers:

```
Authorization: Bearer <token>  
Content-Type: application/json
```

Request Body:

```
{  
  "skillName": "TypeScript",  
  "level": "beginner",  
  "source": "manual"  
}
```

Success Response (201):

```
{  
  "success": true,  
  "message": "Skill added successfully"  
}
```

26. Remove User Skill

Endpoint: DELETE /api/skills/:skillID

Description: Remove skill from user profile

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "success": true,
  "message": "Skill removed successfully"
}
```

⚠ Error Handling

All API errors follow this format:

```
{
  "success": false,
  "error": "Error message here",
  "errorCode": "ERROR_CODE",
  "statusCode": 400
}
```

Common Error Codes

Code	Status	Description
INVALID_CREDENTIALS	401	Wrong email/password
TOKEN_EXPIRED	401	JWT token expired
TOKEN_INVALID	401	Invalid JWT token
UNAUTHORIZED	403	Access denied
NOT_FOUND	404	Resource not found
VALIDATION_ERROR	400	Invalid request data
FILE_TOO_LARGE	400	File exceeds size limit
UNSUPPORTED_FILE	400	Invalid file type
RATE_LIMIT	429	Too many requests
SERVER_ERROR	500	Internal server error

📝 Notes

- All timestamps are in ISO 8601 format (UTC)
 - File uploads limited to 5MB
 - Rate limiting: 100 requests per 15 minutes per IP
 - JWT tokens expire after 7 days
 - AI responses may take 5-10 seconds depending on complexity
-

API Versioning

Current version: v1

Future updates will use versioned endpoints:

- POST /api/v2/career/recommend
-

Support

For API issues:

- Email: api-support@careerpilot.com
 - Docs: <https://docs.careerpilot.com>
-

Last Updated: November 18, 2025