**Programming for Kids**

**Architecture: Model View Controller**

**Front-End: React js**

**Back-End: Django**

**Database: Postgresql**

**User Side**

| **API Endpoint** | **HTTP method** | **Request body** | **Request Response** |
| --- | --- | --- | --- |
| api/signup | POST | {  "type" : 1,  "username" : "user",  "pass" : "user",  “name” : “Shahrukh”,  “email”: “cseshahrukh@gmail.com”  “class” : 7  } | HTTP/1.1 200 OK  {  “token” : ”Successful Sign Up”  } |
| api/login | POST | {  "type" : 1,  "username" : "user",  "pass" : "user",  } | Success Response:   * Status Code: 201 Created * Response Body:   {  "id": 1,  "username": "shahrukh",  "email": "shahrukh@cse.com"  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid user data."  } |
| api/dashboard/newuser | GET |  | * Success Response:   + Status Code: 200 OK   + Response Body:   {  "recommended\_courses": [  {  "id": 1,  "title": "Introduction to Programming",  "description": "A beginner-friendly course to learn the basics of programming.",  "level": "Beginner"  },  {  "id": 2,  "title": "Python for Kids",  "description": "Learn Python programming in a fun and interactive way.",  "level": "Beginner"  },  // Additional recommended courses...  ]  }   * Error Response:   + Status Code: 404 Not Found   + Response Body:   {  "error": "No recommended courses found for new users."  } |
| api/dashboard/incomplete | GET |  | * Success Response:   + Status Code: 200 OK   + Response Body:   {  "incomplete\_courses": [  {  "id": 3,  "title": "JavaScript Basics",  "description": "Continue learning JavaScript and build interactive web applications.",  "level": "Intermediate"  },  // Additional incomplete courses...  ]  }   * Error Response:   + Status Code: 404 Not Found   + Response Body:   {  "error": "No incomplete courses found for the user."  } |
| api/dashboard/completed | GET |  | * Success Response:   + Status Code: 200 OK   + Response Body:   {  "recommended\_courses": [  {  "id": 4,  "title": "Web Development Fundamentals",  "description": "Build a strong foundation in web development and learn HTML, CSS, and JavaScript.",  "level": "Beginner"  },  {  "id": 5,  "title": "Python Web Frameworks",  "description": "Explore popular Python web frameworks like Django and Flask.",  "level": "Intermediate"  },  // Additional recommended courses...  ]  }   * Error Response:   + Status Code: 404 Not Found   + Response Body:   {  "error": "No recommended courses found for completed users."  } |
| /api/curriculum | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "curriculum": [  {  "id": 1,  "title": "Introduction to Programming",  "lessons": [  {  "id": 1,  "title": "Variables and Data Types",  "description": "Learn about variables and different data types in programming."  },  {  "id": 2,  "title": "Conditional Statements",  "description": "Explore conditional statements and decision-making in programming."  },  // Additional lessons...  ]  },  // Additional curriculum sections...  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Curriculum not found."  } |
| api/search-course | GET | {  “keyword”: “python”  } | Success Response:   * Status Code: 200 OK * Response Body:   {  "courses": [  {  "id": 1,  "title": "Python Programming",  "description": "Learn the fundamentals of Python programming language.",  "level": "Beginner",  “class”: 6  },  {  "id": 2,  "title": "Python Data Structure",  "description": "Introduction to Python Data Structure",  "level": "Intermediate",  “class”: 8  },  // Additional courses...  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "No courses found."  } |
| api/courses/{course\_id}/register | POST | {  “course\_id”: “1002”  } | Success Response:   * Status Code: 200 OK * Response Body:   {  "message": "Successfully registered for the course."  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid course registration request."  } |
| api/courses/{course\_id}/progress | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "course\_id": 1,  "progress": 80,  "completed\_lessons": 8,  "total\_lessons": 10  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Course progress not found."  } |
| api/courses/{course\_id}/lesson/{lesson\_id} | GET |  | Request Response:Success Response:   * Status Code: 200 OK * Response Body:   {  "lesson": {  "id": 1,  "title": "Variables and Data Types",  "description": "Learn about variables and different data types in programming.",  "content": "<p>This lesson covers the basics of variables and data types in programming...</p>"  }  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Lesson not found."  } |
| /api/courses/{course\_id}/quiz/{lesson\_id} | GET |  | * Success Response:   + Status Code: 200 OK   + Response Body: json   {  "problems": [  {  "id": 1,  "title": "Sum of Two Numbers",  "description": "Write a program to find the sum of two numbers.",  "type": "coding",  "difficulty": "Easy"  },  {  "id": 2,  "title": "Data Types Quiz",  "description": "Test your knowledge about programming data types.",  "type": "mcq",  "difficulty": "Medium",  "options": [  "A. Integer",  "B. String",  "C. Boolean",  "D. Float"  ]  },  // Additional problems...  ]  }   * Error Response:   + Status Code: 404 Not Found   + Response Body: json   {  "error": "No mcq problems found."  } |
| /api/courses/{course\_id}/quiz/{lesson\_id}/submit | POST | {  "answers": [  {  "question\_id": 1,  "selected\_option": "A"  },  {  "question\_id": 2,  "selected\_option": "C"  },  // Additional answers...  ]  } | Success Response:   * Status Code: 200 OK * Response Body:   {  "score": 80,  "passed": true  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid quiz answers."  } |
| /api/courses/{course\_id}/problems | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "problems": [  {  "id": 1,  "title": "Sum of Two Numbers",  "description": "Write a program to find the sum of two numbers.",  "difficulty": "Easy"  },  {  "id": 2,  "title": "Factorial Calculation",  "description": "Write a program to calculate the factorial of a number.",  "difficulty": "Medium"  },  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "problems not found."  } |
| /api/courses/{course\_id}/problems/{problem\_id} | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "problem\_id": 1,  "title": "Sum of Two Numbers",  "description": "Write a program to find the sum of two numbers.",  "difficulty": "Easy",  "example\_input": "Input: 3, 5",  "example\_output": "Output: 8"  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": “Problem not found."  } |
| /api/courses/{course\_id}/problems/{problem\_id}/submit | POST | {  "solution\_code": "def sum\_of\_two\_numbers(a, b):\n return a + b\n\nresult = sum\_of\_two\_numbers(3, 5)\nprint(result)"  } | * Success Response:   + Status Code: 200 OK   + Response Body: json   {  "message": "Solution submitted successfully.",  “test cases passed”:”8/10”  }   * Error Response:   + Status Code: 400 Bad Request   + Response Body: json   {  "error": "Invalid solution submission data."  } |
| api/capstone/{project\_id} | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "project\_id": 1,  "title": "Build a Personal Portfolio Website",  "description": "Create a personal portfolio website to showcase your programming projects and skills.",  "steps": [  {  "id": 1,  "title": "Step 1: Planning and Wireframing",  "description": "Plan the structure and design of your portfolio website and create wireframes.",  "completed": true  },  {  "id": 2,  "title": "Step 2: Design and Styling",  "description": "Implement the visual design of your portfolio website using HTML and CSS.",  "completed": false  },  // Additional steps...  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Capstone project not found."  } |
| api/capstone/{project\_id}/steps/{step\_id} | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "step\_id": 1,  "title": "Step 1: Planning and Wireframing",  "description": "Plan the structure and design of your portfolio website and create wireframes.",  "completed": true,  "resources": [  {  "id": 1,  "title": "Wireframe Template",  "url": "https://example.com/wireframe-template.pdf"  },  {  "id": 2,  "title": "Portfolio Planning Guide",  "url": "https://example.com/portfolio-planning-guide.pdf"  },  // Additional resources...  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Step details not found."  } |
| api/capstone/{project\_id}/submit | POST | {  "title": "My Capstone Project",  "description": "This is my completed capstone project for the course.",  "repository\_url": "https://github.com/user/repo"  } | Success Response:   * Status Code: 200 OK * Response Body:   {  "message": "Capstone project submitted successfully."  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid capstone project submission data."  } |
| /api/courses/{course\_id}/discussion | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "discussion\_threads": [  {  "id": 1,  "title": "How to declare variables in Python?",  "author": "john\_doe",  "created\_at": "2022-12-15T10:30:00Z",  "replies": [  {  "id": 1,  "author": "jane\_smith",  "content": "In Python, you can declare variables using the assignment operator (=). For example, `name = 'John'` assigns the value 'John' to the variable name.",  "created\_at": "2022-12-15T11:00:00Z"  },  // Additional replies...  ]  },  // Additional discussion threads...  ]  }  Error Response:   * Status Code: 404 Not Found * Response Body:   {  "error": "Discussion threads not found."  } |
| /api/courses/{course\_id}/discussion | POST | {  "title": "How to use loops in JavaScript?",  "content": "I'm having trouble understanding loops in JavaScript. Can someone provide examples?"  } | Success Response:   * Status Code: 201 Created * Response Body:   {  "message": "Discussion thread created successfully."  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid discussion thread data."  } |
| /api/courses/{course\_id}/discussion/{thread\_id} | GET |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "thread\_id": 1,  "title": "How to declare variables in Python?",  "author": "Shahrukh",  "created\_at": "2022-12-15T10:30:00Z",  "content": "I'm new to Python and would like to know how to declare variables. Can anyone help?",  "replies": [  {  "id": 1,  "author": "Prof. Tahmid Hasan",  "content": "In Python, you can declare variables using the assignment operator (=). For example, `name = 'John'` assigns the value 'John' to the variable name.",  "created\_at": "2022-12-15T11:00:00Z"  },  // Additional replies...  ]  }   * Status Code: 404 Not Found * Response Body:   {  "error": "Discussion thread not found."  } |
| api/courses/{course\_id}/discussion/{thread\_id}/reply | POST | {  "content": "Here's an example of a loop in JavaScript: \n\nfor (let i = 0; i < 5; i++) {\n console.log(i);\n}"  } | * Success Response:   + Status Code: 200 OK   + Response Body:   {  "message": "Reply posted successfully."  }   * Error Response:   + Status Code: 400 Bad Request   + Response Body:   {  "error": "Invalid reply data."  } |
| api/student-performance | GET |  | HTTP/1.1 200 OK  {  "student\_id": "12345",  "total\_courses": 5,  "total\_points": 250,  "achievements": [  {  "title": "Bronze Star",  "description": "Earned for completing 3 courses.",  "points": 50  },  {  "title": "Silver Star",  "description": "Earned for completing 5 courses.",  "points": 100  },  {  "title": "Gold Star",  "description": "Earned for achieving a total of 200 points.",  "points": 200  }    ]  } |
| api/recommended course | GET |  | HTTP/1.1 200 OK  {  "courses": [  {  "course\_id": "9876",  "title": "Introduction to Java Programming",  "level": "Beginner",  "duration": “6 weeks”,  "instructor": "Md.Shahrukh Islam"  },  {  "course\_id": "5432",  "title": "Advanced Java Programming",  "level": "Intermediate",  "duration": “7 weeks”,  "instructor": "Sohidul Islam"  },  ]  } |
| api/archived/courses | GET |  | HTTP/1.1 200 OK  {  "courses": [  {  "course\_id": "67890",  "title": "Introduction to C Programming",  "completion\_date": "2023-07-01"  },  {  "course\_id": "54321",  "title": "C Programming(part 2)",  "completion\_date": "2023-06-15"  }  ]  } |

**Teacher Side:**

| **API Endpoint** | **HTTP method** | **Request body** | **Request Response** |
| --- | --- | --- | --- |
| api/teacher/sign-up | POST | {  "type" : 2,  "username" : "teacher",  "pass" : "teacher",  “name” : “Shahrukh”,  “email”: “cseshahrukh@gmail.com”  } | HTTP/1.1 200 OK  {  “token” : ”Successful Sign Up”  } |
| api/teacher/login | POST | {  "type" : 2,  "username" : "user",  "pass" : "user",  } | Success Response:   * Status Code: 201 Created * Response Body:   {  "id": 1,  "username": "shahrukh",  "email": "shahrukh@cse.com"  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid user data."  } |
| api/teacher/courses | POST | {  "title": "Introduction to Programming",  "description": "Learn the basics of programming",  "duration": 6 weeks  } | {  "course\_id": "12345"  } |
| api/teacher/courses/{course\_id}/materials | POST | {  "title": "Programming Basics",  "url": "https://example.com/programming-basics.pdf"  } | {  "message": "Reading material uploaded successfully."  } |
| api/teacher/courses/{course\_id}/quiz-questions | POST | {  "question": "What is a variable?",  "options": ["A. A constant value", "B. A placeholder for storing data", "C. A mathematical operation", "D. A conditional statement"],  "correct\_option": 1  } | {  "message": "Quiz question uploaded successfully."  } |
| api/teacher/courses/{course\_id}/programming-questions | POST | POST /api/courses/12345/programming-questions  {  "question": "Write a program to find the factorial of a number.",  "programming\_language": "Python",  "sample\_input": "5",  "sample\_output": "120"  } | {  "message": "Programming question uploaded successfully."  } |
| api/teacher/courses/{course\_id}/project-questions | POST | POST /api/courses/12345/project-questions  {  "question": "Create a simple calculator program using any programming language."  } | {  "message": "Project question uploaded successfully."  } |
| api/teacher/courses/{course\_id} | PUT | {  "title": "Introduction to Programming (Updated)",  "description": "Learn the basics of programming and algorithms",  "duration": 5 Weeks,  "instructor\_name": "John Doe"  } | {  "message": "Course updated successfully."  } |
| api/teacher/courses/{course\_id}/materials/{material\_id} | PUT | {  "title": "Updated Programming Basics",  "url": "https://example.com/updated-programming-basics.pdf"  } | {  "message": "Reading material updated successfully."  } |
| api/teacher/courses/{course\_id}/quiz-questions/{question\_id} | PUT | {  "question": "What is a variable? (Updated)",  "options": ["A. A constant value", "B. A placeholder for storing data", "C. A mathematical operation", "D. A conditional statement"],  "correct\_option": 2  } | {  "message": "Quiz question updated successfully."  } |
| api/teacher/courses/{course\_id}/programming-questions/{question\_id} | PUT | {  "question": "Write a program to find the factorial of a number. (Updated)",  "programming\_language": "Python",  "sample\_input": "6",  "sample\_output": "720"  } | {  "message": "Programming question updated successfully."  } |

**Admin Side:**

| **API Endpoint** | **HTTP method** | **Request body** | **Request Response** |
| --- | --- | --- | --- |
| api/admin/login | POST | {  "type" : 3,  "username" : "admin",  "pass" : "admin",  } | HTTP/1.1 200 OK  {  “token” : ”login successful”  } |
| api/admin/approve-teacher | POST | {  "user\_id": 12345  } | * Success Response:   + Status Code: 200 OK   + Response Body:   {  "message": "Teacher approved successfully."  }   * Error Response:   + Status Code: 400 Bad Request   + Response Body:   {  "error": "Invalid teacher approval request."  } |
| api/admin/courses/{course\_id}/verify | PUT | PUT /api/admin/courses/12345/verify | {  "message": "Course verified and approved successfully."  } |