**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: json   {  "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: json   {  "error": "No recommended courses found for new users."  } |
| api/dashboard/incomplete | GET |  | * Success Response:   + Status Code: 200 OK   + Response Body: json   {  "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: json   {  "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 |  | 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}/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/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}/complete | POST |  | Success Response:   * Status Code: 200 OK * Response Body:   {  "message": "Step completed successfully."  }  Error Response:   * Status Code: 400 Bad Request * Response Body:   {  "error": "Invalid step completion request."  } |
| 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: json   {  "message": "Reply posted successfully."  }   * Error Response:   + Status Code: 400 Bad Request   + Response Body: json   {  "error": "Invalid reply data."  } |
| api/uploadPDF | POST | {  “mode”: “formdata”,  “formdata”: [ {  “key” : ”file”,  “src” : ”C:\\Users\\filename.pdf”,  “type” : ”file”  }]  } | HTTP/1.1 201 OK  {  “msg” : “Upload successful”  } |
| api/validate | POST | {  “form\_id” : 12345  } | HTTP/1.1 200 OK  {  “image”: “encoded\_image”  "form\_id": "12345",  "time" : "12:12:12",  "date" : "2019-01-01",  "workspace\_id" : "456789",  "fields" : [  {  "field\_label": "Name",  "field\_type": "whole\_text",  "field\_text": "Mehedi Hasan",  "mistake": "false",  "suggestions": []  },  {  "field\_label": "Name in letters",  "field\_type": "letter\_by\_letter",  "field\_text": [  "M",  "e",  "h",  "e",  "d",  "i",  " ",  "H",  "a",  "s",  "a",  "n"  ],  "mistake": "false",  "suggestions": []  },  {  "field\_label": "Birth certificate number",  "field\_type": "letter\_by\_letter",  "field\_text": [  "1",  "8",  "0",  "4",  "0",  "6",  "7",  "8"  ],  "mistake": "false",  "suggestions": []  },  {  "field\_label": "birth date",  "field\_type": "letter\_by\_letter",  "field\_text": [  "1",  "8",  "0",  "4",  "1",  "9",  "9",  "9"  ],  "mistake": "false",  "suggestions": []  },  {  "field\_label" : "District",  "field\_type" : "whole\_text",  "field\_text" : "Dheka",  "mistake" : "true",  "suggestions" : [  "Dhaka"  ]  },  {  "field\_label": "Gender",  "field\_type": "checkbox",  "checked\_index": "0",  "mistake": "false"  },  {  "field\_label": "Religion",  "field\_type": "checkbox",  "checked\_index": "-1",  "mistake": "true"  },  {  "field\_label": "class",  "field\_type": "checkbox",  "checked\_index": "8",  "mistake": "false"  },  {  "field\_label": "roll",  "field\_type": "letter\_by\_letter",  "field\_text": [  "0",  "1",  "2"  ],  "mistake": "false",  "suggestions": []  },  {  "field\_label": "marital status",  "field\_type": "checkbox",  "checked\_index": "1",  "mistake": "false"  }  ]  } |
| api/drafts/create | POST | edited fields same as validate response | HTTP/1.1 201 OK  {  “form\_id” : 123  } |
| api/drafts/info | POST | {  “draft\_id” : 123,  \*\*  other fields are same as createDrafts  } | Same as api/validate |
| api/drafts/update | POST | {  “draft\_id” : 123,  } | HTTP/1.1 201 OK  {  “msg” = “updated”  } |
| api/drafts/list | GET | {  } | HTTP/1.1 200 OK  {  {  “form\_id” : 123,  “name” : “Mehedi Hasan”,  “time” : ,  “date” : ,  “class” : 7  }  {  }  } |
| api/drafts/my | GET | {  } | HTTP/1.1 200 OK  {  {  “form\_id” : 123,  “name” : “Mehedi Hasan”,  “time” : ,  “date” : ,  “class” : 7  }  {  }  } |
| api/history/list | GET | {  } | HTTP/1.1 200 OK  {  {  “form\_id” : 1234,  “name” : “Mehedi”,  “class” : 7,  “date” : ,  “time” : ,  “user\_id” : “admin”  }  } |
| api/history/create | POST | edited fields same as validate response | HTTP/1.1 201 OK  {  “form\_id” : 1234  } |
| api/history/info | POST | {  “form\_id” : 1234,    } | Same as api/validate |
| api/history/update | POST | {  “form\_id” : 1234,  \*\*  other fields are same as History/Create  } | HTTP/1.1 201 OK  {  “msg” = “history updated successfully”  } |
| 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": "John Smith"  },  {  "course\_id": "5432",  "title": "Advanced Java Programming",  "level": "Intermediate",  "duration": “7 weeks”,  "instructor": "Jane Doe"  },  ]  } |

**Teacher Side:**

| **API Endpoint** | **HTTP method** | **Request body** | **Request Response** |
| --- | --- | --- | --- |
|  |  |  |  |

**Admin Site:**

| **API Endpoint** | **HTTP method** | **Request body** | **Request Response** |
| --- | --- | --- | --- |
| api/login | POST | {  "type" : 2,  "username" : "admin",  "pass" : "admin",  } | HTTP/1.1 200 OK  {  “token” : ”Amsvdg  AsbDhdaksHdbdabdYa4684aDw48wa4d6sa4d8Wa8w4d68ad”  } |
| api/users | GET | {  } | HTTP/1.1 200 OK  {  {  “user\_id” : “123”  “user\_name” : “sanju”,  “email” : “[sanju@gmail.com](mailto:sanju@gmail.com)”,  }  {  . . .  }  } |
| api/users/info | POST | {  “user\_id” : “123”  } | HTTP/1.1 200 OK  {  “user\_id” : “123”  “user\_name” : “sanju”,  “email” : “[sanju@gmail.com](mailto:sanju@gmail.com)”,  “pass” : “\*\*\*\*\*\*\*\*”  “mobile” : “01812121212”  } |
| api/users/update | POST | {  “user\_id” : “123”  “user\_name” : “sanju”,  “email” : “[sanju05@gmail.com](mailto:sanju@gmail.com)”,  “pass” : “\*\*\*\*\*\*\*\*\*”  “mobile” : “01812121212”  } | HTTP/1.1 201 OK  {  “msg” : “user info updated  successfully”  } |
| api/users/create | POST | {  “user\_name” : “sanju”,  “email” : “[sanju05@gmail.com](mailto:sanju@gmail.com)”,  “pass” : “\*\*\*\*\*\*\*\*”  “mobile” : “01812121212”  } | HTTP/1.1 201 OK  {  “msg” : “user created  successfully”  } |
| **WORKSPACE** | | | |
| api/workspace/create | GET | {} | HTTP/1.1 200 OK |
| api/workspace/create | POST | {  "name": "class 2 section B",  "class": 2,  "section": "B",  "year": "2020",  "min roll": 1,  "max roll": 60,  "total students": 60,  "description":""  } | HTTP/1.1 201 OK  {  “msg” : “workspace created  successfully”  } |
| api/workspace/list | GET | {} | HTTP/1.1 200 OK  {  "workspaces": [  {  "name": "class 2 section A",  "id": "1",  "class": 2,  "section": "A",  "year": "2020",  "pre\_validation": 20,  "validated": 10,  "total uploaded": 20,  "total students": 30,  "out of date": false  },  {  "name": "class 2 section B",  "id": "2",  "class": 2,  "section": "B",  "year": "2020",  "pre\_validation": 13,  "validated": 3,  "total uploaded": 24,  "total students": 60, “ “out of date": false  }  ]  } |
| api/workspace/:workspaceid | GET | {  "id": "1"  } | HTTP/1.1 200 OK  {  "name": "class 2 section A",  "id": "1",  "class": 2,  "section": "A",  "year": "2020",  "pre\_validation": 20,  "validated": 10,  "total uploaded": 20,  "total students": 30,  "out of date": false,  "validated rolls" : [  {  "roll": 1,  "form id": 100  },  {  "roll": 2,  "form id": 101  }  ],  "non validated" : [  {  "date": "2020-01-01",  "time": "10:00:00",  "student name": "razin",  "form id": 1000  },  {  "date": "2020-01-01",  "time": "10:00:00",  "student name": "azrin",  "form id": 1001  }  ]  } |
| api/workspace/:workspaceid | DELETE | {  "id": "1"  } | HTTP/1.1 200 OK |
| api/workspace/:workspaceid | PUT | {  "id": "1",  } | HTTP/1.1 200 OK |
| api/workspace/:workspaceid/validate/:formid | GET | {  "workspace id": "1",  "form id": "100"  } | Same as validate |