Cairo University Racing Team

Season 2026

Software Development Task

Hello Applicant,

Reading this task means that you have successfully passed the selection phase as one phase out of three. So Welcome to the second phase which is the following Task.

Please read these notes carefully before you begin the Task, and you **MUST** stick to them.

Notes:

Due to the tight deadline, we will consider your efforts, finish as much as you can, try to make it appealing.

1. Searching Online is **Allowed**. However, using a code that you don't understand is a problem.
2. Make your code Well Organized, Commented and Clean as much as you can.

- The deadline for the task will be due to **29th Aug 2025 11:59 pm.** Late Delivery will not be Accepted.

- If you have any Question, feel free to send an email to sda.curt@gmail.com with a subject of "**Software Task Inquiry**". Please Stick to the Subject name and we will reply as soon as possible.

Project Management System Task

Develop a MERN (MongoDB, Express.js, React, Node.js) stack application that implements a CRUD (Create, Read, Update, Delete) operation involving two related database tables. The application should allow users to manage a simple project management system. The task duration is 4 days, and it is acceptable if the entire task is not completed within this period.

1.2 Functional Requirements

1.2.1 Authentication

- The system should have User Authentication with username and password.

1.2.2 CRUD Operations

- **Domain**: A project management system where users can manage **Projects** and **Tasks**.
- **Database Collections**:
  1. **Projects**:
     – Fields: _id, title (string, required), description (string), createdAt (timestamp).
  2. **Tasks**:
     – Fields: _id, projectId (reference to Project _id), title (string, re-quired), status (enum: ["ToDo", "InProgress", "Done"]), createdAt (timestamp).
- **CRUDoperations**:
  – **Create**: Users can create projects and tasks (linked to a project).
  – **Read**: List all projects and tasks for a specific project. Display details for individual projects/tasks.
  – **Update**: Allow updating project title/description and task title/status.
  – **Delete**: Allow deletion of projects (cascading to delete related tasks) and tasks.

1.2.3 Technical Requirements

- **Backend**:

  – Use Node.js with Express.js for the server.

  – MongoDB with Mongoose for database operations.

  – Implement RESTful API endpoints for CRUD operations. –

  Example endpoints:

  ∗ POST /api/project - Create a project.

  ∗ GET /api/project - List all projects.

  ∗ PUT /api/project/:id - Update a project.

  ∗ DELETE /api/project/:id - Delete a project.

  ∗ Similar endpoints for tasks (/api/task).

  – Use environment variables.

- **Frontend**:

  – Use React with React Router for navigation. – Use

  Axios or Fetch for API calls.

  – Implement a simple UI with Tailwind CSS for styling.

  – Pages:

  ▪ Login Page.

  ▪ Sign up Page.

  ▪ Page to list projects.

  ▪ Project detail page to list tasks.

  ▪ Task detail page to view task details.

- **Database**:

  – MongoDB with Mongoose schemas for Project and Task. –

  Establish relationships using ObjectId references.

  – Implement cascading deletes (e.g., deleting a project removes its tasks).

1.2.4 Deliverables

- **Backend**:
  - Fully functional Express.js server with RESTful API.
  - MongoDB database migrations (generated by mongoose).
  - Environment configuration (.env file example).
- **Frontend**:
  - React application with routing.
  - Responsive design.

1.2.5 Tools and Technologies

- **Frontend**: React, React Router, Axios, Tailwind,
- **Backend**: Node.js, Express.js, MongoDB, Mongoose.
- **Development**: MongoDB Compass (for easier database management).
- **Version Control**: Git/GitHub for code management.

Delivery Requirements

1. Record a video with a maximum length of 5 minutes explaining your task. Upload this video on Google Drive and make sure it's public. The video name must be "<Your name> SDA Task Solution".
2. Link to a GitHub repository that contains your complete source code
3. Send an email to sda.curt@gmail.com that contains the above links with the subject: "**SDA Task Solution**".

Good Luck,

Amr Hany Head of Team

Omar Nabil Vice Head of Team