UCS2611 – INTERNET PROGRAMMING LABORATORY

MINI PROJECT

INTERNTRACKER FULL STACK APPLICATION

Kavin T 3122 22 5001 058

Manishkumar S 3122 22 5001 070

Mohamed Imran Fareeth S 312222 5001 072

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

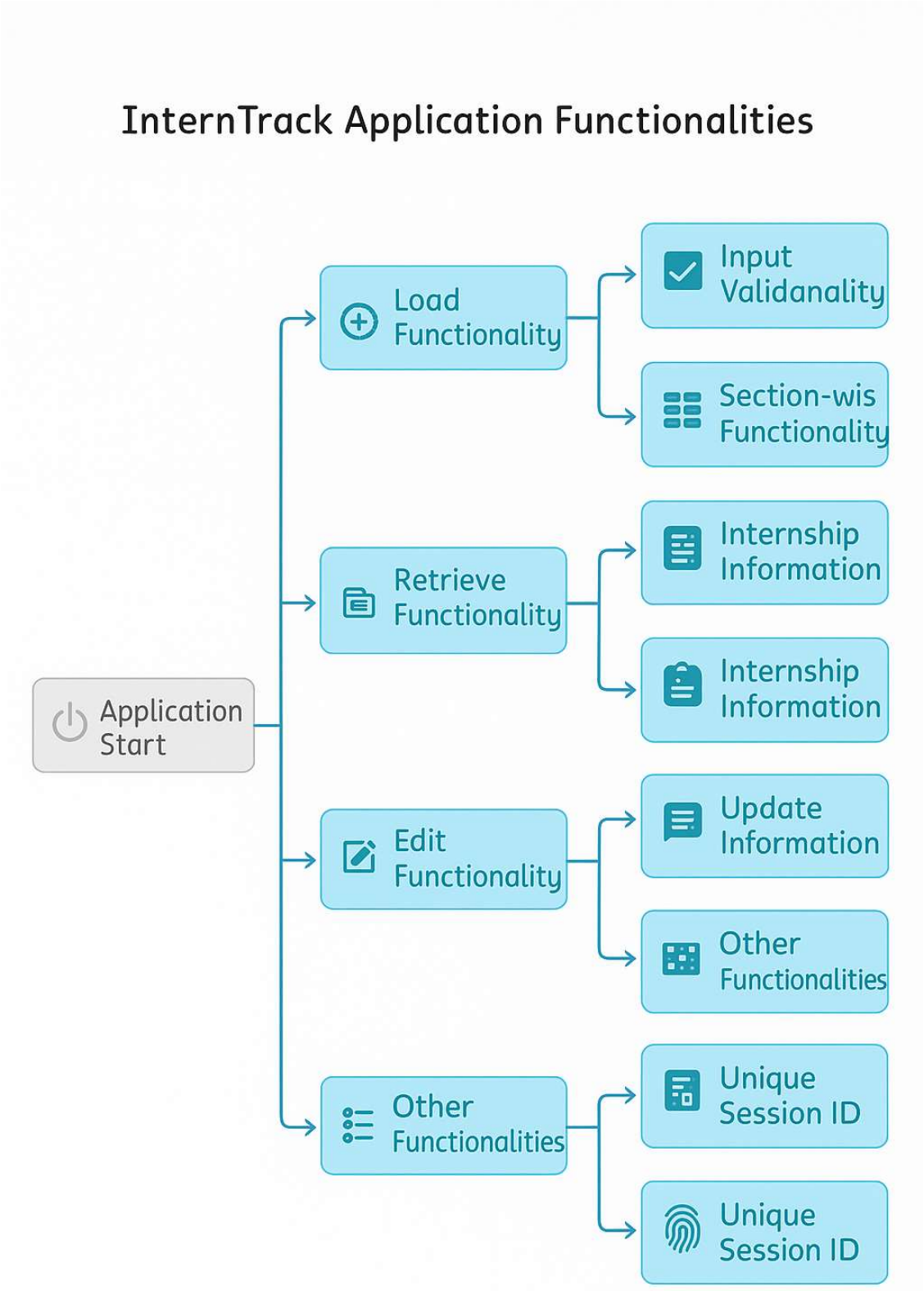
(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

Intern Tracker Project

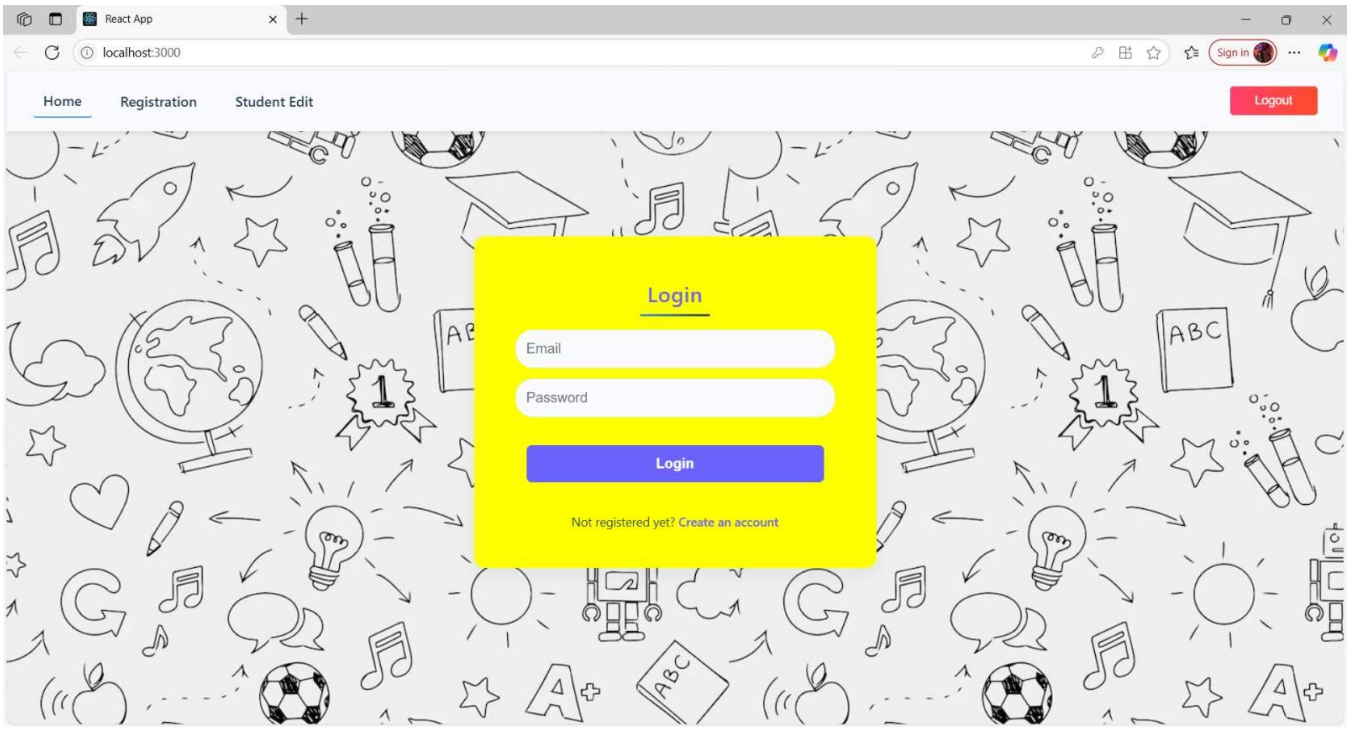
- InternTrack is a web-based system designed to streamline the process of collecting, managing, and tracking internship details of students.
- The system supports both student and coordinator roles, offering a centralized platform to record internship data, upload relevant documents, and communicate with coordinators.
- The backend is built using Node.js, Express.js, and MongoDB, with Mongoose for schema handling and Multer for file uploads.

Functionalities implemented:



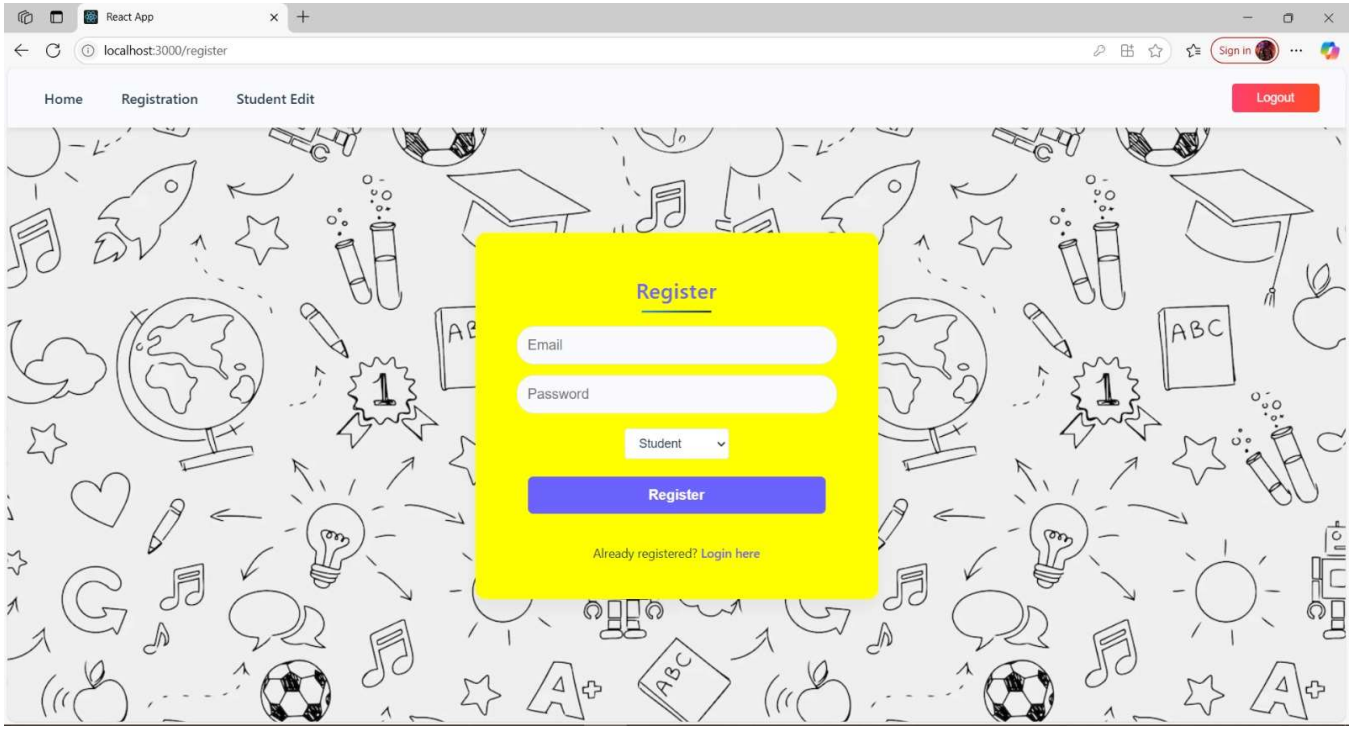
This is the website interface

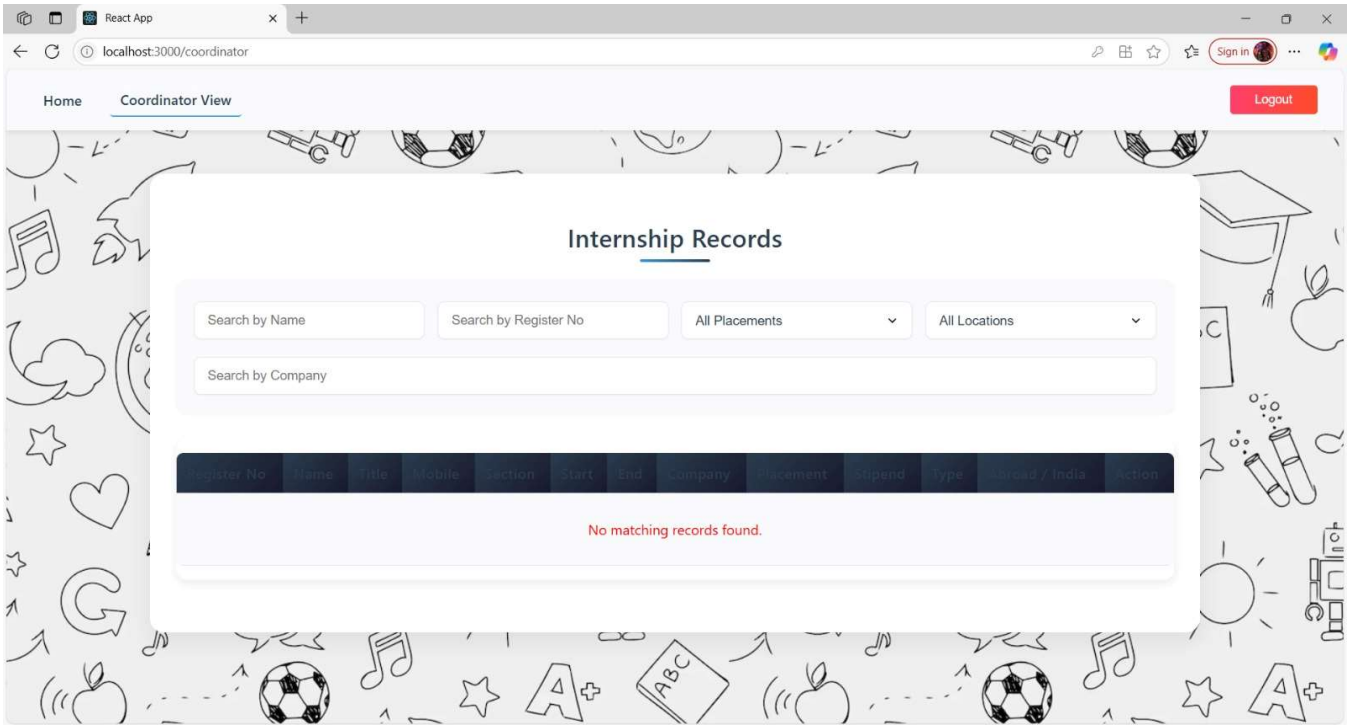
Login:



Authentication of user can be done using this login functionality

Register:





Coordinator View - Internship Records Dashboard

The image above showcases the **Coordinator View** of the InternTrack application, a web-based portal developed to manage and monitor internship data of students efficiently.

This view is specifically designed for coordinators to access and filter internship records submitted by students. The core functionalities of this page include:

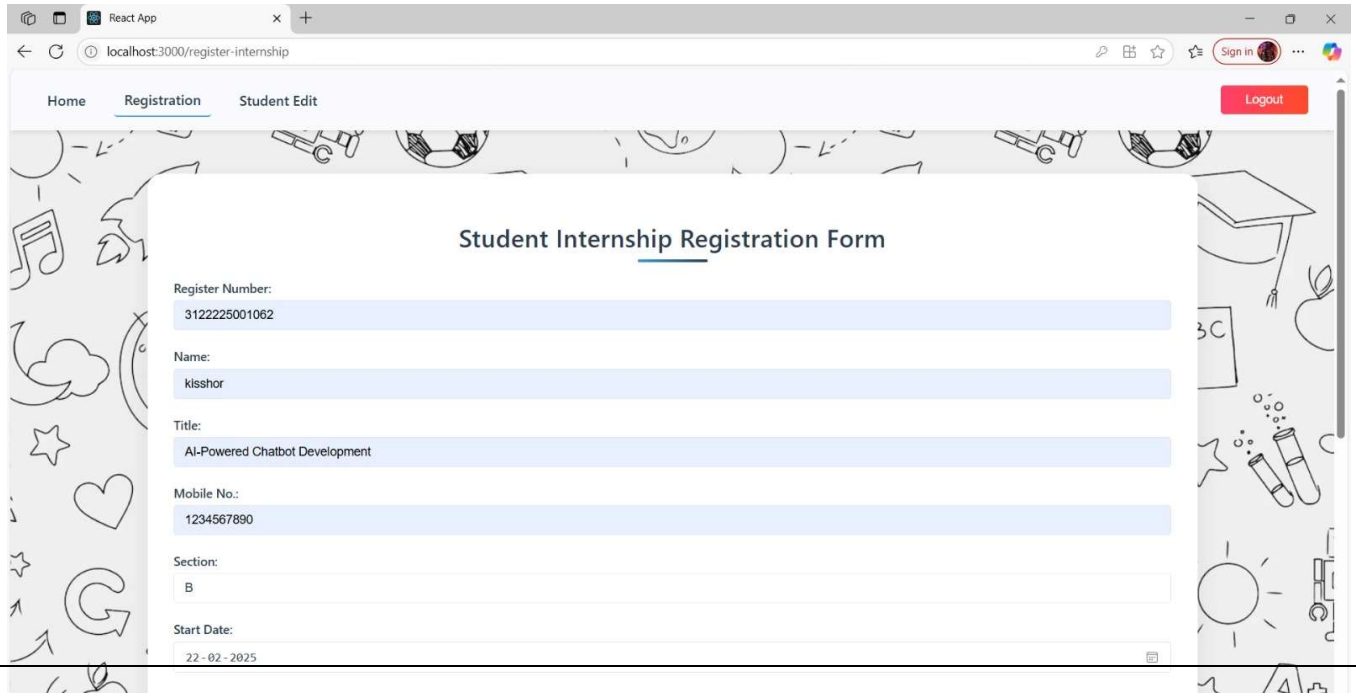
- **Search Functionality**:
  Coordinators can perform a quick search using student name, register number, or company name.

- **Filter Options**:
  Dropdown filters are provided to narrow down records based on:
  - **Placement Status** (e.g., placed, not placed)
  - **Location** (e.g., India, Abroad)

- **Data Table View**:
  A responsive and structured table displays key internship details like:
  - Register Number
  - Name
  - Title
  - Mobile
  - Section
  - Internship Start and End Dates
  - Company Name
  - Placement Info
  - Stipend
  - Internship Type (In-office, Remote, etc.)
  - Location (India/Abroad)
  - Action Buttons for Editing or Verifying

- **Empty State Handling**:
  As seen in the image, when no matching records are found, a user-friendly message "No matching records found." is displayed in red, ensuring clear communication.

This view enhances administrative workflow by offering a centralized and interactive interface for reviewing and managing internship data seamlessly.



Student Internship Registration Form:

The above screenshot displays the **Student Internship Registration Form** page of the InternTrack application. This interface allows students to submit their internship details for verification and approval by the institution.

Key Features:

- **Input Fields for Core Details**:

Students are required to fill out key fields such as:

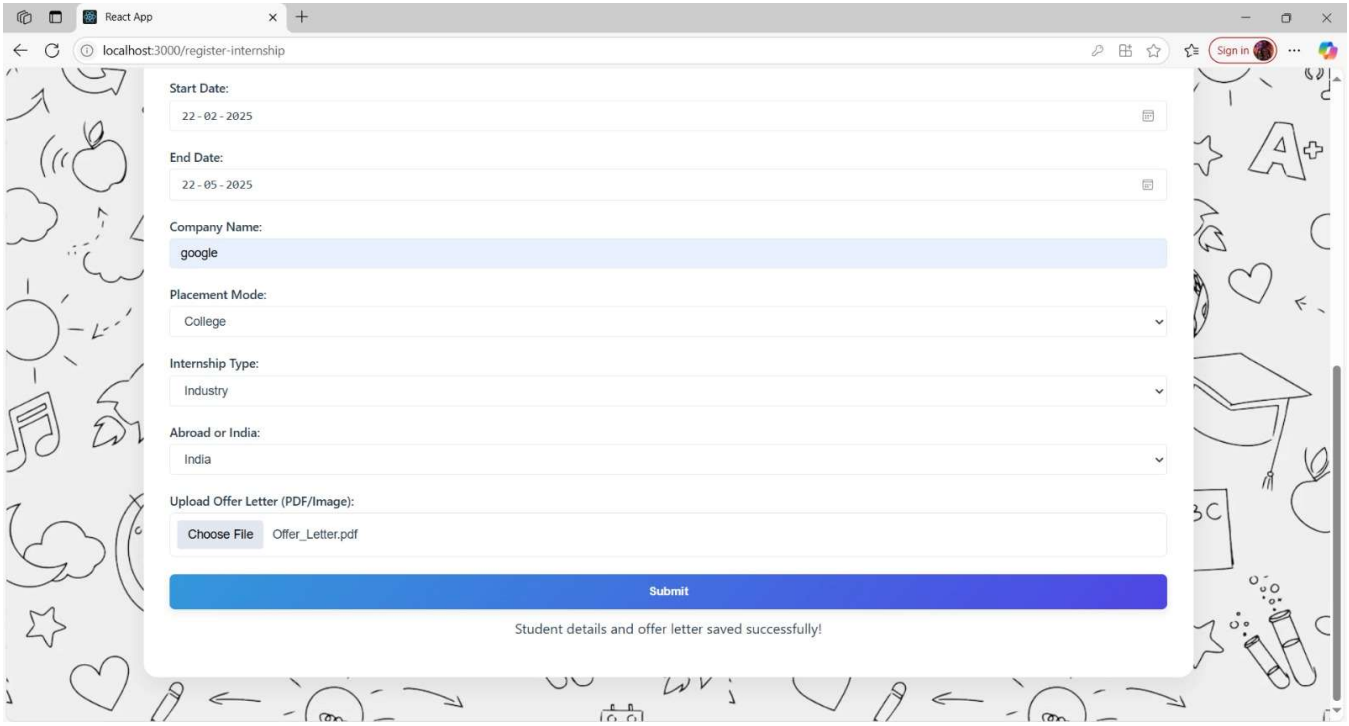
  - **Register Number** (example shown: 3122225001062)
  - **Name** (example: kisshor)
  - **Internship Title** (e.g., *AI-Powered Chatbot Development*)
  - **Mobile Number**
  - **Section** (e.g., B)

- **Start Date Picker**:
  The form includes a user-friendly date picker component for selecting the **internship start date**, reducing the chance of input errors.

- **UI Design**:
  The form is built with a clean and minimal UI, using read-only inputs to display registered student information, and editable fields for new inputs. It sits on a visually engaging background that maintains an academic theme.

- **Navigation Bar**:
  The top navigation provides quick access to:
  - **Home**
  - **Registration**
  - **Student Edit**

A **Logout** button ensures secure session handling.
This form acts as the entry point in the InternTrack system for students to record and submit their internship details, streamlining the approval and tracking process.



Student Internship Registration Form (Extended View)

This screenshot represents the **extended section** of the **Student Internship Registration Form** within the InternTrack application. It completes the form submission process by collecting detailed information about the internship and verifying authenticity through document upload.

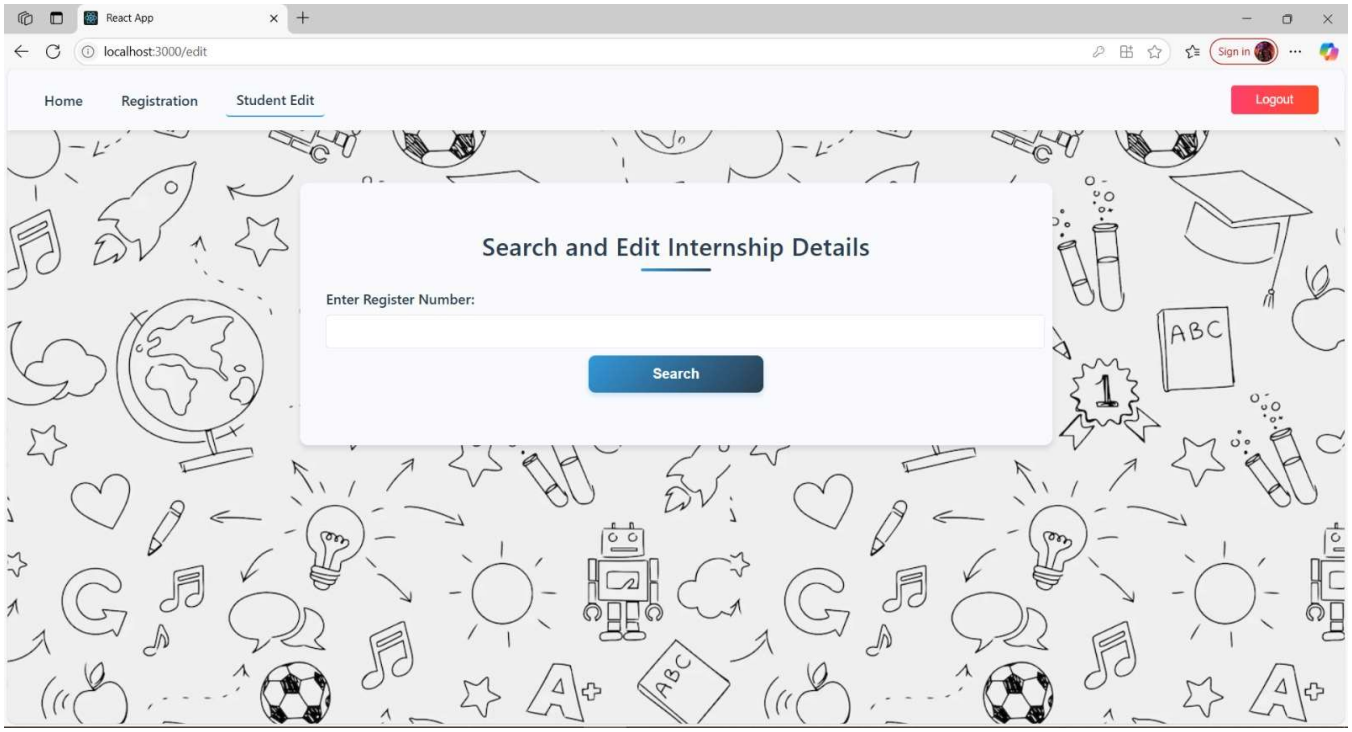
Additional Fields Captured:

- **End Date**:
  Allows the student to specify the internship's conclusion using a date picker component.
- **Company Name**:
  The name of the company offering the internship is recorded (e.g., *Google* in the example).
- **Placement Mode**:
  A dropdown menu enables students to select how the internship was secured (e.g., *College*, *Self-arranged*, etc.).
- **Internship Type**:
  Defines the nature of the internship (e.g., *Industry*, *Research*, etc.).
- **Abroad or India**:

Specifies the internship location — whether it's domestic or international.

- **Upload Offer Letter**:
  A file upload component is provided to attach an official internship offer letter (PDF or image format). This ensures authenticity and allows the institution to verify the internship claim.
- **Submission Confirmation**:
  Upon successful form submission, a success message is shown — **"Student details and offer letter saved successfully!"** — confirming the data has been stored.
- **Submit Button**:
  A visually distinct gradient button is used for submission, enhancing UI clarity and accessibility.

This part of the form finalizes the registration process and plays a vital role in verifying and recording legitimate internship details for institutional approval.

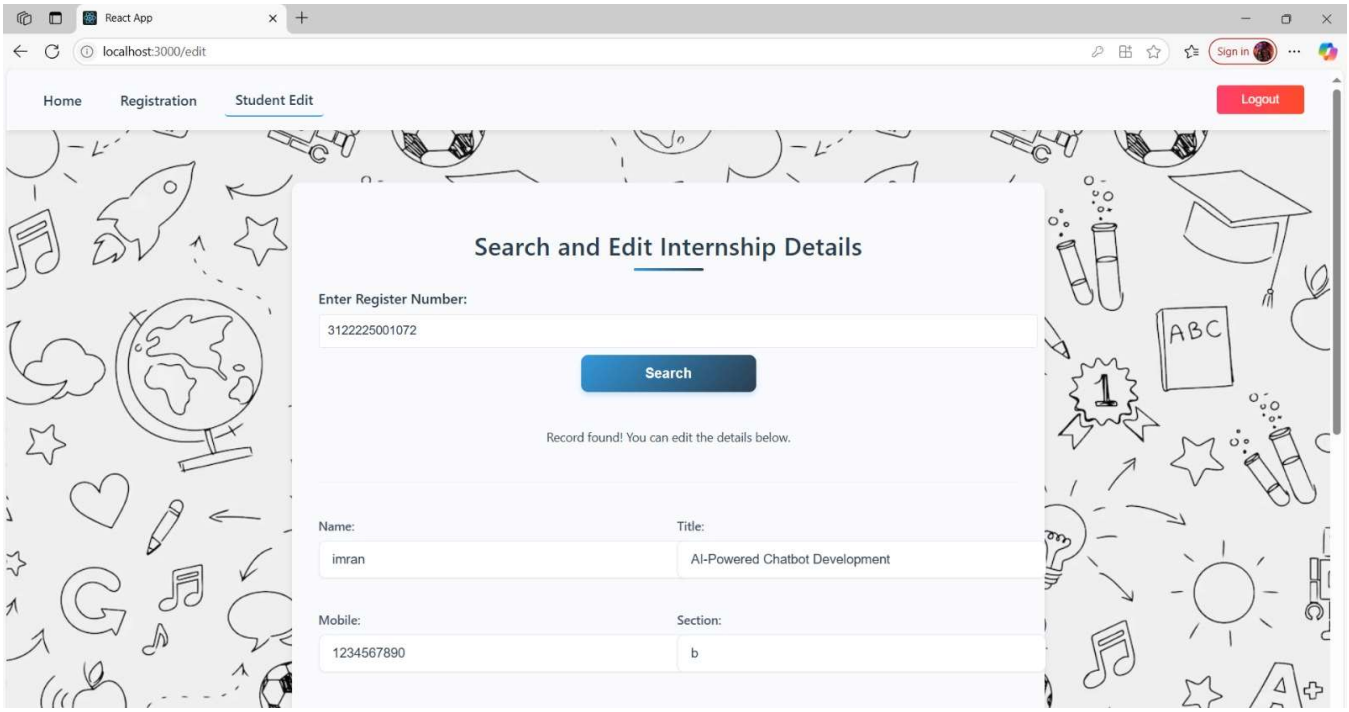


Search and Edit Internship Details:
This screenshot illustrates the **Search and Edit** module of the InternTrack system, designed to allow users (primarily students or administrators) to **retrieve and update previously submitted internship records**.

Key Features:

- **Route**:
  The URL localhost:3000/edit indicates that this is the Edit page within the application.
- **Title Section**:
  The title *"Search and Edit Internship Details"* is clearly displayed with styling similar to the registration form, ensuring UI consistency.
- **Input Field – Register Number**:
  Users are prompted to enter a **Register Number** to search for a student's internship record. This ensures that editing access is tied to a unique identifier.
- **Search Button**:
  A clean, gradient-styled **Search** button is provided to trigger the fetch operation. Upon clicking, the system queries the database and displays the student's saved details in an editable format.

Purpose:
This page serves as a **central hub for editing existing internship information**. It supports data correction, offer letter re-uploads, internship date updates, and more — promoting flexibility and data accuracy.



Record Retrieval and Editing Functionality:

This page demonstrates the **result of a successful search query** within the *Search and Edit Internship Details* section.

Key Features:

- **Register Number Search**:
  The register number 3122225001072 has been entered and submitted via the **Search** button.
- **Confirmation Message**:
  A feedback message "Record found! You can edit the details below." confirms that a valid entry was located in the database.
- **Editable Fields**:
  Upon retrieval, the system renders editable fields with the intern's data:
  - **Name**: imran
  - **Title**: AI-Powered Chatbot Development
  - **Mobile**: 1234567890
  - **Section**: b

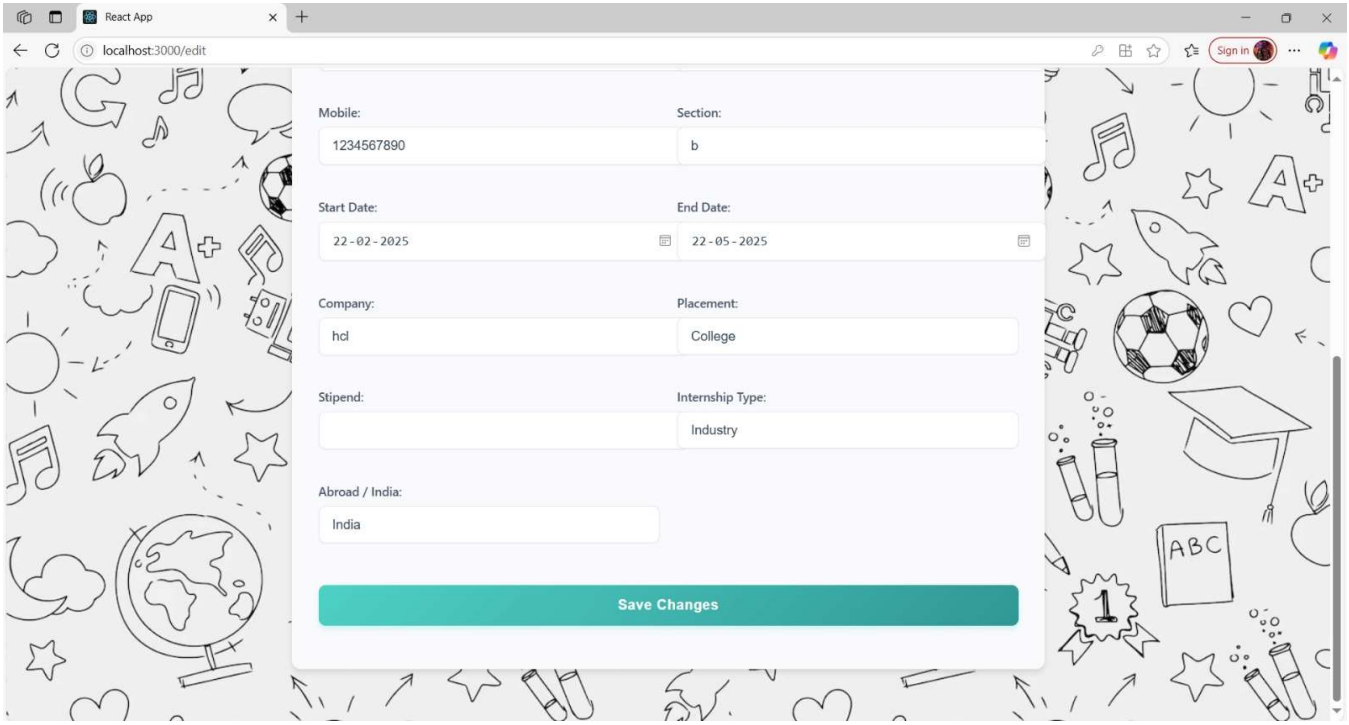
These values are now available for modification, allowing users to correct or update any outdated information.

Purpose:

This UI segment enhances the system's usability by:

- Enabling **dynamic data retrieval**
- Supporting **inline editing of student internship details**
- Ensuring **data consistency and accuracy**

It acts as the final step before updating the database with revised information.



Final Editable Internship Form (Before Save)

This form allows the admin or user to **modify complete internship details** for a selected student.

Key Editable Fields Shown:

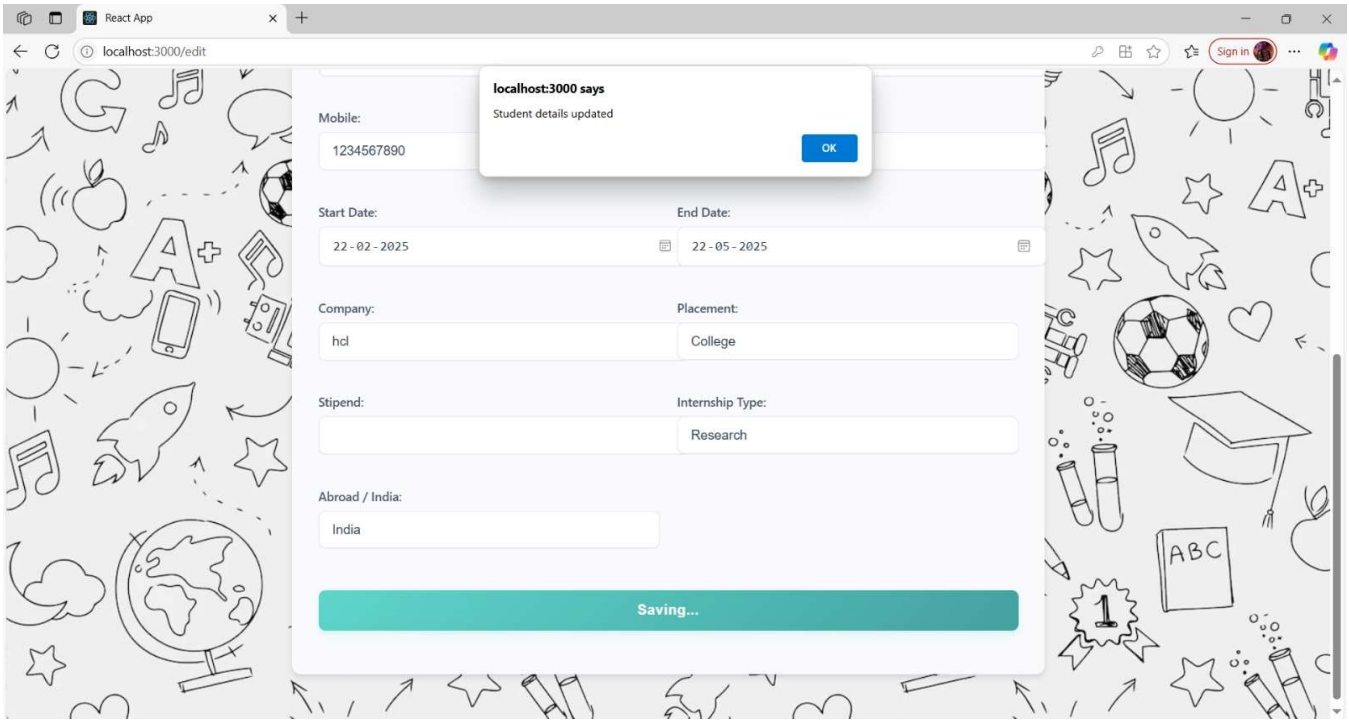
- **Start Date**: 22-02-2025
- **End Date**: 22-05-2025
- **Company**: hcl
- **Placement**: College
- **Stipend**: (Currently empty – likely optional or awaiting input)
- **Internship Type**: Industry
- **Abroad / India**: India

Final Action:

- A clearly visible **Save Changes** button is provided at the bottom to submit the updated data.

Functional Notes:

- This is the **complete edit interface**, combining previously shown basic details (like name, section, and mobile) with advanced internship info.
- Data will be saved through an API (most likely a PUT or PATCH request) to update the backend database once **Save Changes** is clicked.



Edit Internship Details Page

1)Application Context: This screenshot represents the **Edit Internship Details** section of a React-based web application. It is used for updating internship information of students. The application runs on localhost:3000, indicating that it's in the development environment.

2. Key UI Components:

- **Form Fields:**
  - **Mobile:** The student's mobile number (1234567890).
  - **Start Date / End Date:** Represents the internship duration (22-02-2025 to 22-05-2025). These fields use a date picker component.
  - **Company:** The name of the company offering the internship (HCL).
  - **Placement:** Specifies the nature of the internship placement (College).
  - **Stipend:** Currently left blank, implying either an unpaid internship or data yet to be filled.
  - **Internship Type:** Specifies the category or focus of the internship (Research).
  - **Abroad / India:** Indicates the internship's geographic location (India).
- **Action Button:**
  - A button labeled Saving... is visible, which suggests that the form submission is in progress or has just completed.
- **Feedback Mechanism:**
  - A browser alert box (localhost:3000 says) pops up to inform the user: **"Student details updated"**, indicating that the form submission was successful.

3. UI Aesthetics:

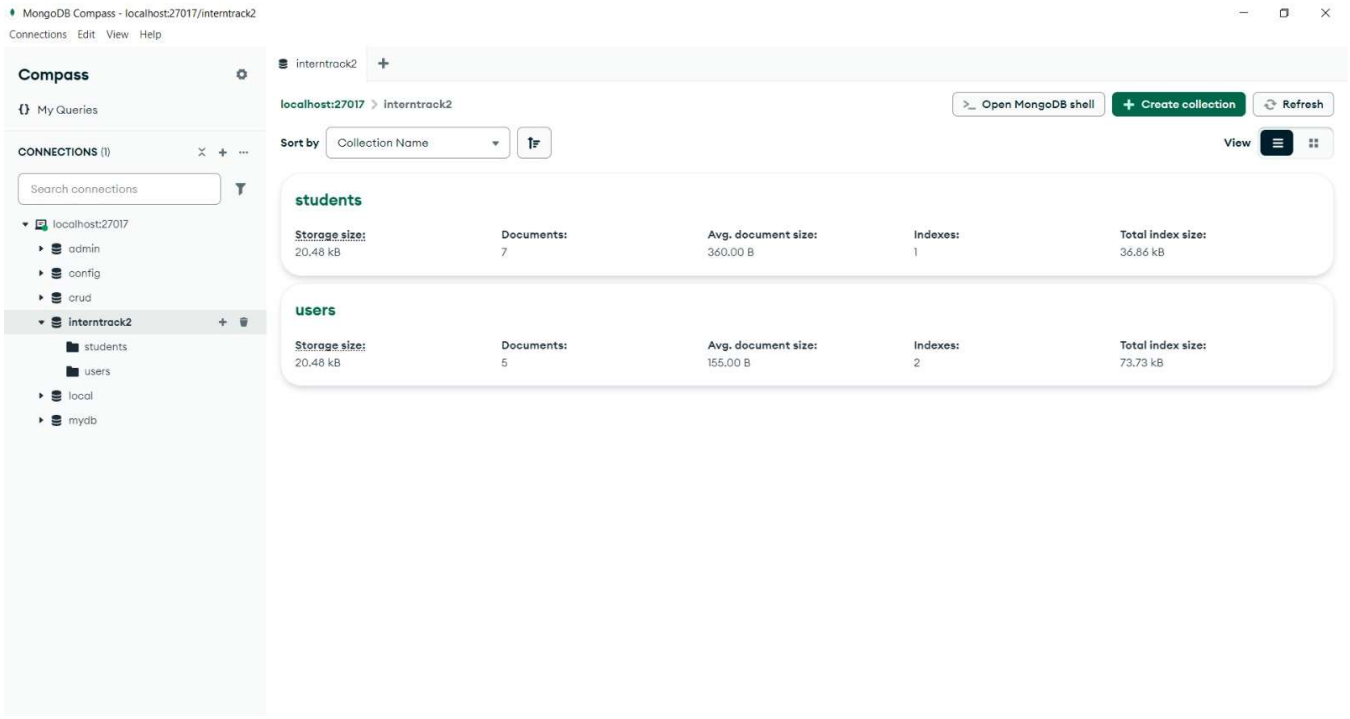
- The background consists of an educational theme with icons representing books, graduation caps, sports, and music, enhancing the student-focused design.
- The submit button has a gradient effect, adding a modern touch to the UI.

4. Functional Flow:

- User updates the necessary fields.
- Upon clicking the submit button, the form triggers an update event (likely using a PUT or POST request to the backend).
- After a successful update, the application uses alert() to confirm the update to the user.

5. Technology Stack (Inferred):

- **Frontend:** React.js
- **State Management / Form Handling:** Likely using React Hooks or Formik.
- **Backend:** Not visible here, but typically includes an API to receive and store the data.
- **Date Picker:** A custom or third-party React date picker component.



MongoDB Compass – Database View

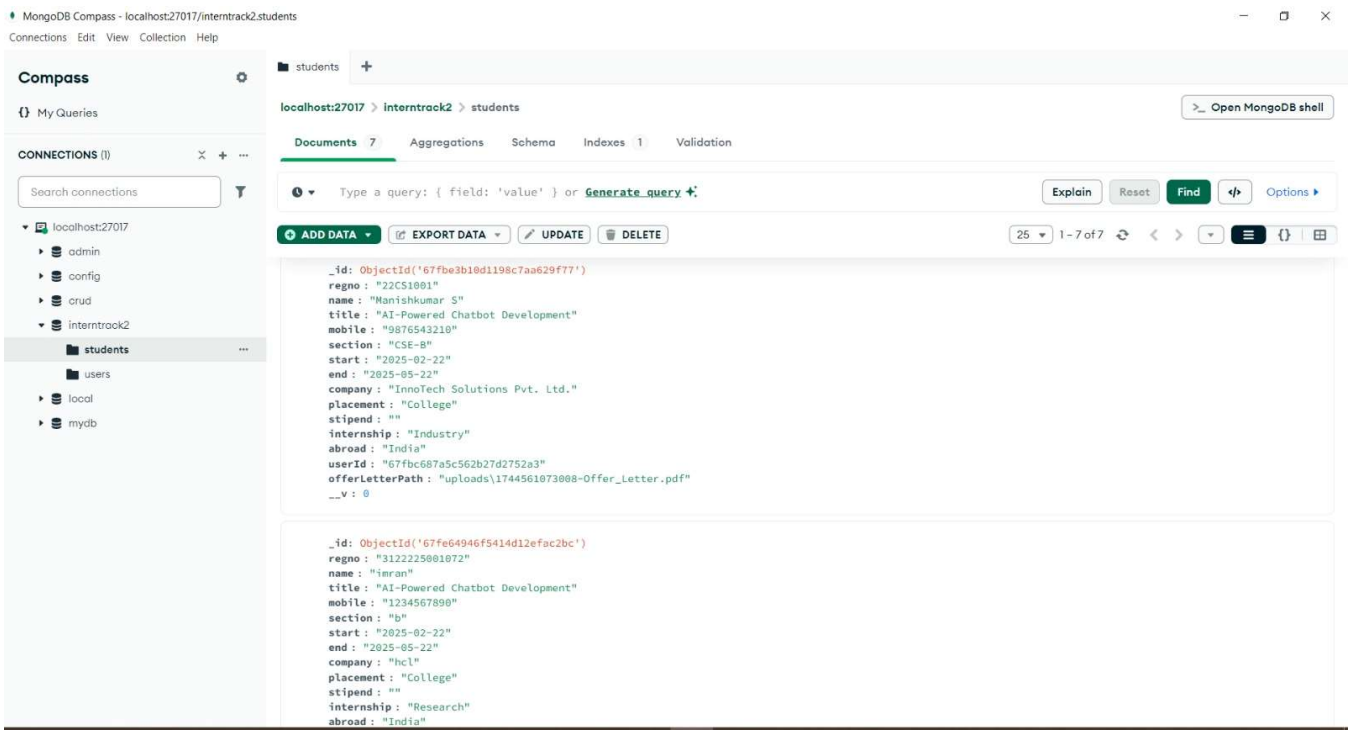
This screenshot displays the structure of the interntrack2 database in **MongoDB Compass**, which is used for managing student internship data.

Database: interntrack2

This database contains two collections:

- **students** – Stores details related to student internships such as mobile number, start and end dates, company name, internship type, placement info, and location.
- **users** – Stores user-related data, such as login credentials or profile information required for authentication and access control within the application.

The collections are accessed and updated by the React frontend (as seen in the previous screenshot). MongoDB Compass is used here to visually manage the data and verify the records.



This report examines the "students" collection within the "internstock2" database, as visualized using MongoDB Compass. The database is hosted locally on a MongoDB server at localhost:27017. The "students" collection currently contains 7 documents, each representing a student and their internship details.

Each student document follows a consistent structure, including the following key fields:

- _id: A unique identifier for each student document (ObjectId).
- regno: The student's registration number (String).
- name: The student's name (String).
- title: The title of the student's internship (String).
- mobile: The student's mobile phone number (String).
- section: The student's academic section (String).
- start: The start date of the internship (ISODate).
- end: The end date of the internship (ISODate).
- company: The company where the student is undertaking the internship (String).
- placement: The source of the placement (String).
- stipend: The stipend amount (String, currently empty in the displayed records).
- internship: The type or domain of the internship (String).
- abroad: The location of the internship (String).
- userId: A reference to a user in another collection (ObjectId), likely linking student records to user accounts.
- _v: A versioning field (Number).
- offerLetterPath: The file path to the student's offer letter (String, present in some records).

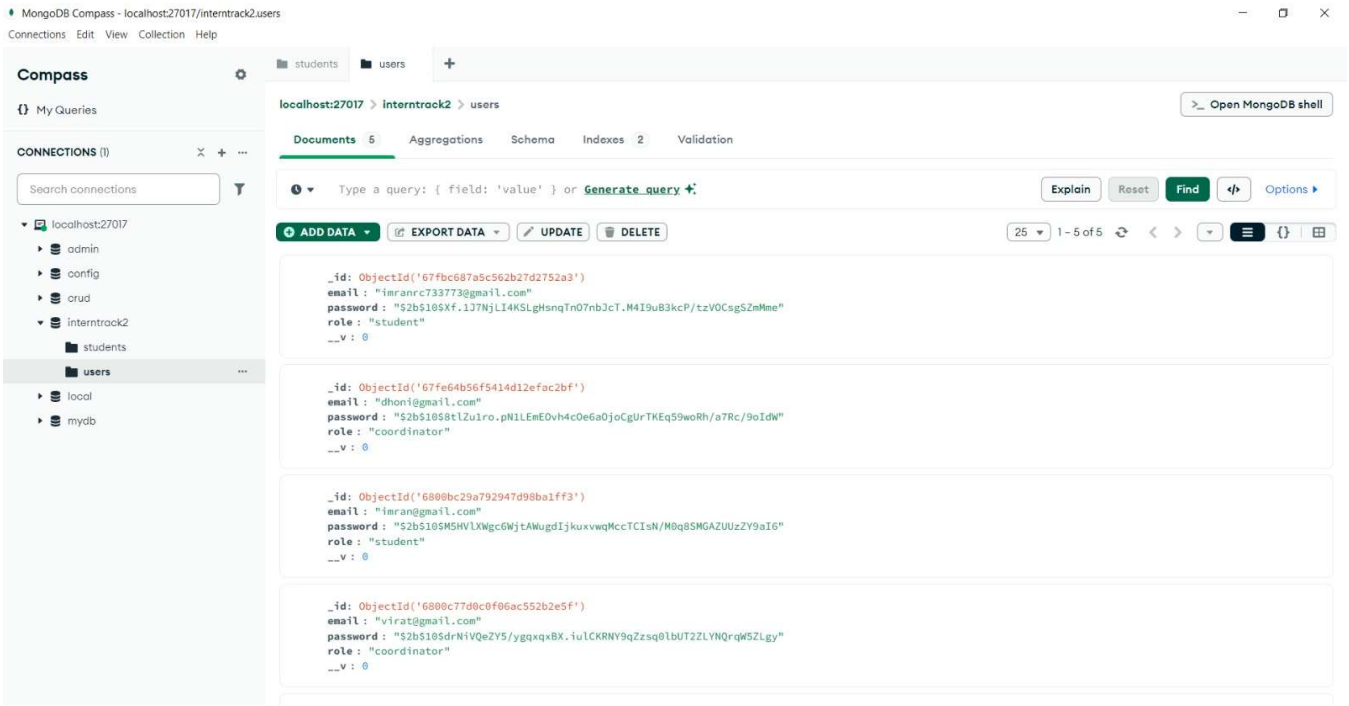
Sample Data Analysis:
Two sample student documents are currently displayed:
Student 1:

- regno: "22531801"
- name: "Manishkumar S"
- title: "AI-Powered Chatbot Development"
- mobile: "9361501230"
- section: "CSE-B"
- start: ISODate("2025-02-22T00:00:00Z")
- end: ISODate("2025-05-22T00:00:00Z")
- company: "InnoTech Solutions Pvt. Ltd."
- placement: "College"
- stipend: ""
- internship: "Industry"
- abroad: "India"
- userId: ObjectId("607fcb75c362b27d27752a3")
- offerLetterPath: "uploads/1714496167368-Offer_Letter.pdf"
- _v: 0

Student 2:

- regno: "212225001072"
- name: "Irfan"
- title: "AI-Powered Chatbot Development"
- mobile: "7234567890"
- section: "B"
- start: ISODate("2025-02-22T00:00:00Z")
- end: ISODate("2025-05-22T00:00:00Z")
- company: "HCl"
- placement: "College"
- stipend: ""
- internship: "Research"
- abroad: "India"
- userId: ObjectId("607fcb849e5f41d1e9fa2bc5")
- _v: 0

Both students are undertaking an internship titled "AI-Powered Chatbot Development" with the same start and end dates (February 22nd, 2025 to May 22nd, 2025) and secured their placement through the college in India. However, they differ in their registration numbers, names, mobile numbers, academic sections, the companies they are interning at, and the type of internship (Industry vs. Research). Only the first student's record includes a path to their offer letter. The stipend field is currently empty for both displayed records. The presence of the userId field suggests a potential relationship with a "users" collection, enabling the linking of student internship data to individual user accounts within the system.
In conclusion, the "students" collection serves to store detailed information about students and their respective internship placements. Further analysis of the complete dataset could reveal more comprehensive insights into the internship program.



This report examines the "users" collection within the "internstock2" database, as visualized using MongoDB Compass. This database resides on a local MongoDB server accessible at localhost:27017. The "users" collection currently holds 5 documents, each representing a user within the system.
Each user document adheres to a consistent schema, containing the following key fields:

- _id: A unique identifier for each user document (ObjectId).
- email: The email address of the user (String).
- password: A hashed or encrypted representation of the user's password (String).
- role: The role or privilege level assigned to the user (String). The observed roles

are "student" and "coordinator".
- _v: A versioning field (Number).
-

Sample Data Analysis:

The screenshot displays the first five documents within the "users" collection, illustrating the structure and the different roles present: "student" and "coordinator". The password fields contain hashed values, indicating secure storage.

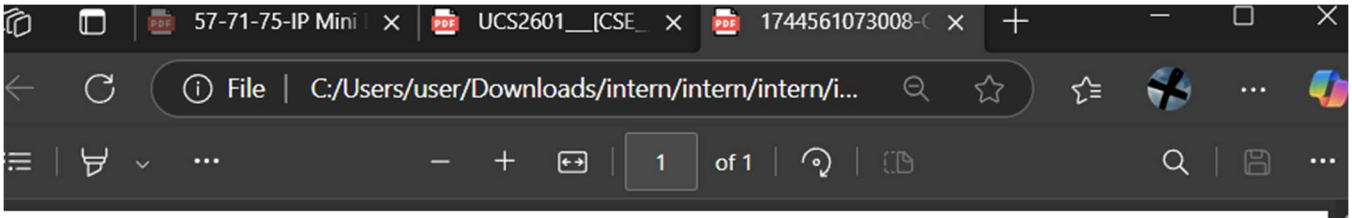
Observations:

- The "users" collection manages user information for the system.
- Users are assigned roles, with the displayed data showing "student" and "coordinator" roles, suggesting different levels of access or functionality.
- Passwords are stored securely using hashing.
- Each user has a unique identifier (_id) and an associated email address.

Potential Relationships:

The "students" collection, as previously examined, contains a userId field. This field likely references the _id of a document in the "users" collection, establishing a relationship between student records and their corresponding user accounts.

In summary, the "users" collection is responsible for storing user information, including email, hashed passwords, and roles ("student" or "coordinator"). This collection likely plays a crucial role in user authentication and authorization within the application, and it is probably linked to the "students" collection through the userId field.



InnoTech Solutions Pvt. Ltd

Internship Offer Letter

Date: 13 April 2025

Dear Manishkumar S,

We are pleased to offer you an internship position at **InnoTech Solutions Pvt. Ltd.** As part of our collaboration with your academic institution. Below are the details of your internship:

- **Register Number:** 22CS1001
- **Section:** CSE-B
- **Mobile No.:** 9876543210
- **Company Name:** InnoTech Solutions Pvt. Ltd.
- **Title:** AI-Powered Chatbot Development
- **Internship Type:** Industry
- **Location of Internship:** India
- **Internship offered through college**
- **You will receive a stipend of Rs. 8000**
- The internship will **commence on 10 March 2025** and **conclude on 10 June 2025**.

Please confirm your acceptance by replying to this letter.

Sincerely,

HR Department

InnoTech Solutions Pvt. Ltd.

Conclusion:

The InternTrack Full Stack Web Application effectively addresses the need for a streamlined and efficient system to manage internship data within academic institutions. By leveraging modern web technologies like the MERN stack and incorporating features such as role-based access, dynamic data handling, and document management, the application ensures a user-friendly and scalable platform for both students and coordinators.

Through various functionalities including data loading, retrieval, viewing, and editing, InternTrack simplifies the process of internship tracking while ensuring data integrity and accuracy. The removal of unnecessary manual efforts, along with automation of core processes, helps coordinators focus on evaluation and guidance rather than administrative tasks.

Overall, InternTrack enhances transparency, accountability, and collaboration in the internship process, making it a valuable asset for academic institutions seeking to digitize and optimize student internship management.

Learning Outcome:

1. Implemented RESTful APIs to handle CRUD operations, including retrieving, updating, and securing internship-related data.
2. Built a full-stack web application using the MERN stack, effectively integrating MongoDB, Express.js and Node.js.
3. Developed the ability to design and maintain back-end databases that can handle document uploads and maintain unique records for each student.
4. Learnt to implement user authentication and authorization mechanisms, ensuring secure access for different roles such as students and coordinators.
5. Implemented security best practices such as input validation, enhancing the robustness of the web application.