



App Class.Com

A student management Website

Shakib ,Al Amin
Ridoan & Mizan
06/13/2023

Guided By:

Abdullah Al Mahfuj

Guest Instructor
Computer Technology



Submitted By:



Shakib Hossain Readoy

5th Semester ,1st Shift

Roll No: 579455

Reg No: 150 214 80 87

Al Amin

5th Semester ,1st Shift
Roll No: 579388388
Reg No: 150 214 81541 54





Habibur Rahman Mizan

5th Semester ,1st Shift

Roll No: 57932432

Reg No: 150 214 81 10



Ridoan

5th Semester ,1st Shift

Roll No: 579404

Reg No: 150 214 81 38

Department of Computer Technology

MoulviBazar Polytechnic Institute

Session: 2020-21

Index:

Tropic.	Paze No
1. Introduction of the web.	4
2. Required Equipment	4
3. Front End Design	5
4. React Components	6
5. Front End Output	6
6. Backend Development	8
7. Node,Express,Mongoose	8
8. Backend Output	9

Introduction of the site

এই Website টি মূলত একটি School এর Students দের Data নেওয়ার জন্য তৈরি করা হয়েছে। এখানে School এর মূল Website এর সাথে একটি Register form যুক্ত করা হয়েছে। যার মাধ্যমে শিক্ষার্থীরা তাদের তথ্য প্রেরণ করতে পারবে। এরপর প্রেরিত তথ্য Website এর Databases এ জমা হবে। এরপর Database এর তথ্য Website এর Dashboard এ দেখা যাবে। এবং Dashboard এর Data , update & DeleteDelete করা যাবে।

Required Equipment

এটি একটি MERN Stack project. অর্থাৎ এখানে JavaScript এবং JavaScript এর Library & Framework সমূহ ব্যবহার করা হয়েছে।

এখানে Front end এ ব্যবহার করা হয়েছে,

1. Tailwind Css,
2. DaisyUi,
3. React js.
4. React Router dom@6

এছাড়া Bakend এ ব্যবহার করা হয়েছে,

1. Node js
2. Express js
3. Mongoose.

এছাড়াও Surver Running রাখতে Nodemon package এবং, MongoDBMongoDb এর Graphical user interface হিসেবে MongodbMongodb compass ব্যবহার করা হয়েছে। Package গুলো Install করার জন্য Npm (Node Package Manager) ব্যবহার করা হয়েছে।

Front end Design

Tailwind Css

Custom Custom css এ অধিক Class লেখার দরকার হয়, এতে Class Name ভাবার প্রয়োজন হয়। Bootstrap এ এই সমস্যার সমাধান থাকলেও Bootstrap এ Customization Customization এর সুবিধা থাকে না। এ কারণে এই Project এ Tailwind Css ব্যবহার করা হয়েছে।

DaisyUi

Tailwind Css এর সাথে Themes এবং Extra Features পাওয়ার জন্য DaisyUi ব্যবহার করা হয়েছে।

React Js

Framework হিসেবে এখানে React js ব্যবহার করা হয়েছে। React এ আলাদা আলাদা কম্পোনেন্ট তৈরি করে একটি ফুল ওয়েবসাইট তৈরি করতে হয়। বর্তমানে সবচেয়ে বেশি ব্যবহৃত ফেমারগুলোর মধ্যে রিয়াক্ট অন্যতম। এখানে সরাসরি es6 (ecmascript-6) ব্যবহারের সুবিধা রয়েছে।

React Router dom@6

এবং ওয়েবের Loading টাইম কমানোর জন্য React Router dom@6 ব্যবহার করা হয়েছে। React Router dom@6 React Router এর সর্বশেষ ভাস্বন। এখানে <Link> বা <NavLink> tag এর মাধ্যমে <a> এর default behaviour change করা যায়। এতে রাউট পরিবর্তনের সময় সাইবার রিলোড হয় না।

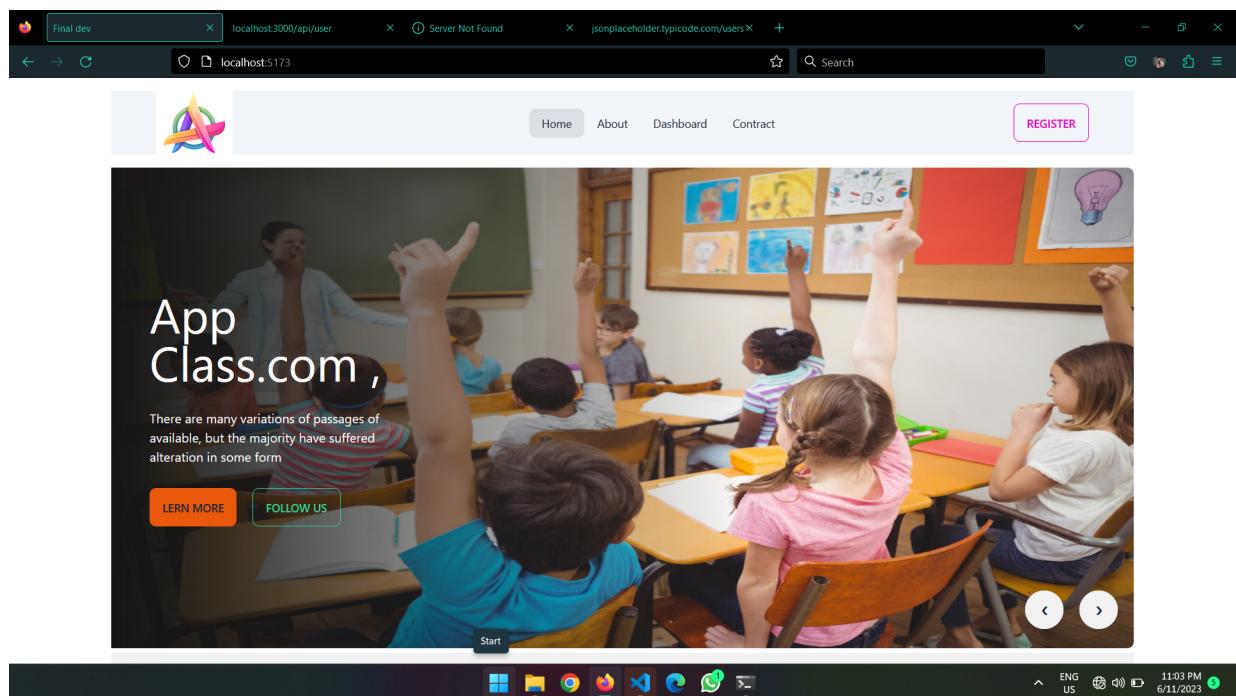
React Components

React এ কম্পনেন্ট তৈরির জন্য .jsx extension ব্যবহার করা হয়। এবং এর মাধ্যমে যারা JavaScript এর যাবতীয় ফাংশনালিটিস এর সাথে Html code ও ব্যবহার করা যায়। এছাড়া fetch api বা Axios ও jsx file এ লিখা যায়।

Front end output

সম্পূর্ণ FrontEnd ডিজাইন করার পর আমাদের প্রজেক্টের ফ্রন্ট পেজের আউটপুট নিচের চিত্রের মত পাওয়া যায়।

Home page:



Dashboard:

1
Leanne Graham
Bret
Sincere@april.biz

2
Ervin Howell
Antonette
Shanna@melissa.tv

UPDATE DELETE

UPDATE DELETE

Final dev | localhost:3000/api/user | jsonplaceholder.typicode.com/users | docs - Google Search | localhost:5173/dashboard | Search | Home About Dashboard Contact REGISTER

2:25 AM ENG US 6/12/2023

Input From:

FIRST NAME
Name

LAST NAME
Title

DEPARTMENT
Department

ROLL NO
Roll No

SHIFT
First

SECTION
A/B

SUBMIT

Final dev | localhost:3000/api/user | Server Not Found | jsonplaceholder.typicode.com/users | localhost:5173/register | Search | Home About Dashboard Contact REGISTER

11:03 PM ENG US 6/11/2023

Backend Development

Node Js

JavaScript এর বিভিন্ন প্যাকেজ ব্যবহার করে এই প্রজেক্টটি সম্পন্ন করা হয়েছে। এক্ষেত্রে Npm (Node Package Manager) ব্যবহার করা হয়েছে।

Express Js

Backend এর রাউটার হিসেবে Express server ব্যবহার করা হয়েছে। বর্তমানে ব্যাক এন্ড সার্ভারগুলোর মধ্যে অন্যতম Express Js. Express এর সাহায্যে Mongodb এর Crude অপারেশন গুলো সম্পন্ন করা হয়েছে।

Mongoose

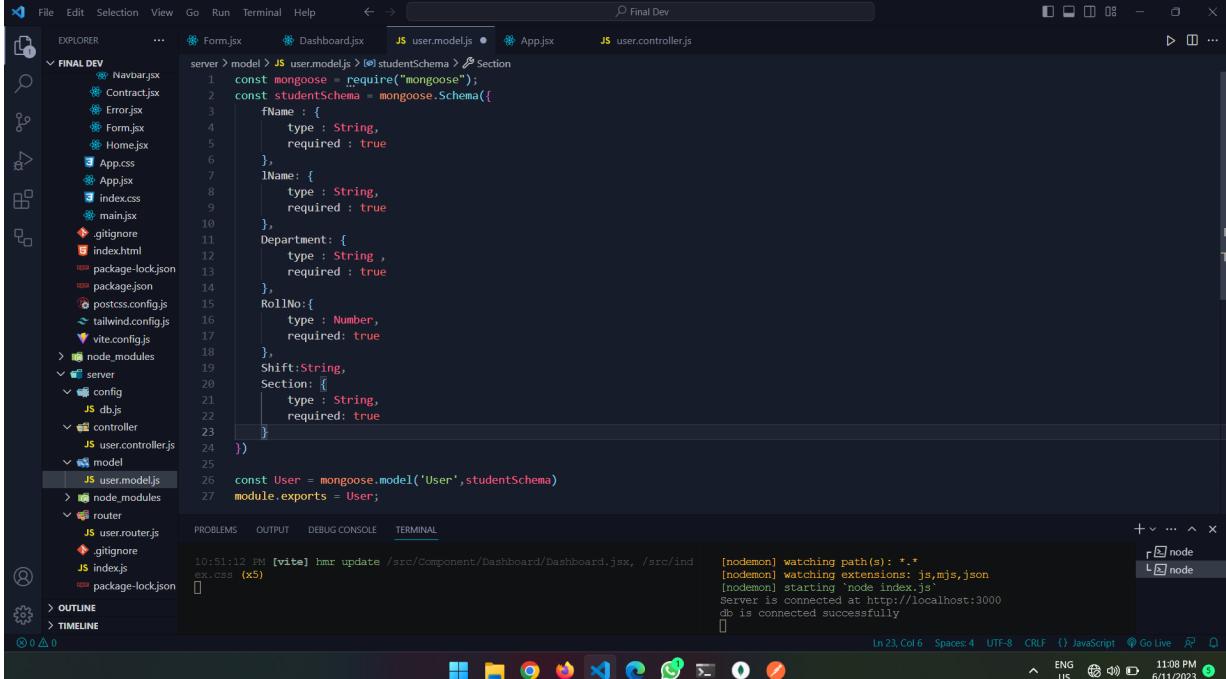
MongoDb এর অধিক জটিলতা এরোতে Mongoose Package টি ব্যবহার করা হয়েছে। এতে অধিক কোড লেখার প্রয়োজন হয় না।

MongoDb Crud Operation

Crud operation এর মাধ্যমে মূলত ফর্ম থেকে ডাটা ডাটাবেজে আনা হয়েছে এবং Axios ব্যবহার করে ডাটা গুলো ড্যাশবোর্ডে প্রদর্শন করা হয়েছে। Dashboard থেকে Crud operation ব্যবহার করে ডাটা আপডেট বা ডিলিট করা যায়।

Backend Output

Input Schema with validation:



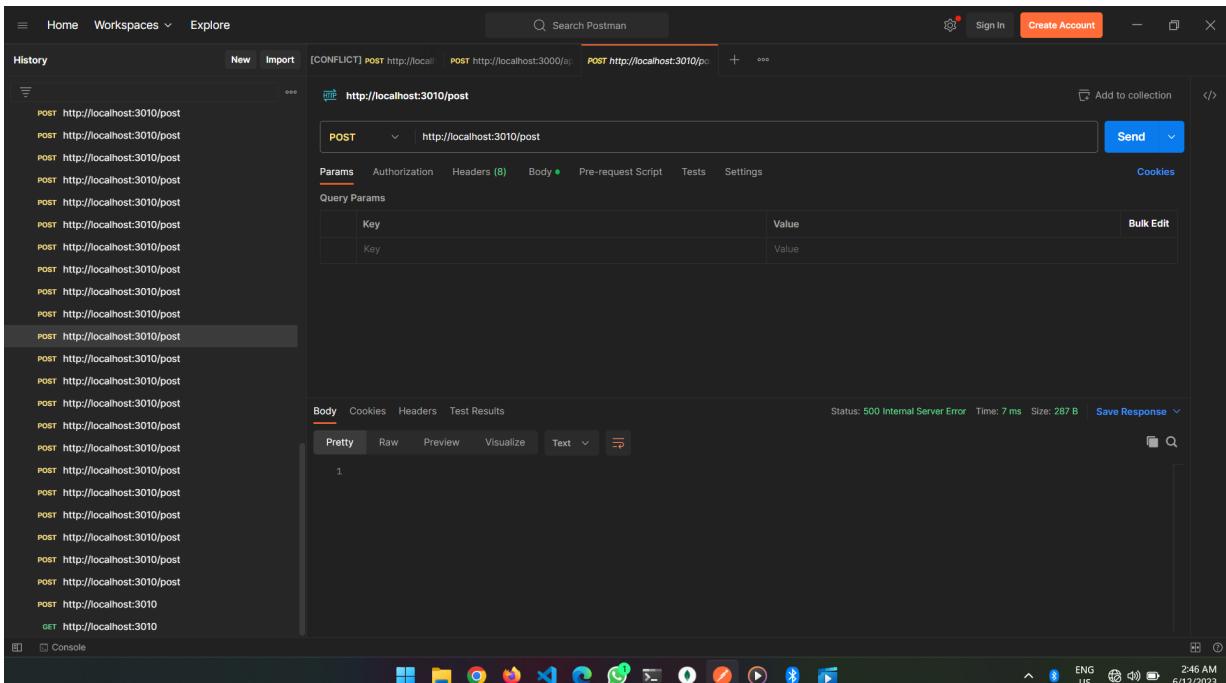
```

const mongoose = require("mongoose");
const studentSchema = mongoose.Schema({
  fName: {
    type: String,
    required: true
  },
  lName: {
    type: String,
    required: true
  },
  Department: {
    type: String,
    required: true
  },
  RollNo: {
    type: Number,
    required: true
  },
  Shift: String,
  Section: {
    type: String,
    required: true
  }
});

const User = mongoose.model('User', studentSchema);
module.exports = User;

```

Input with post men:

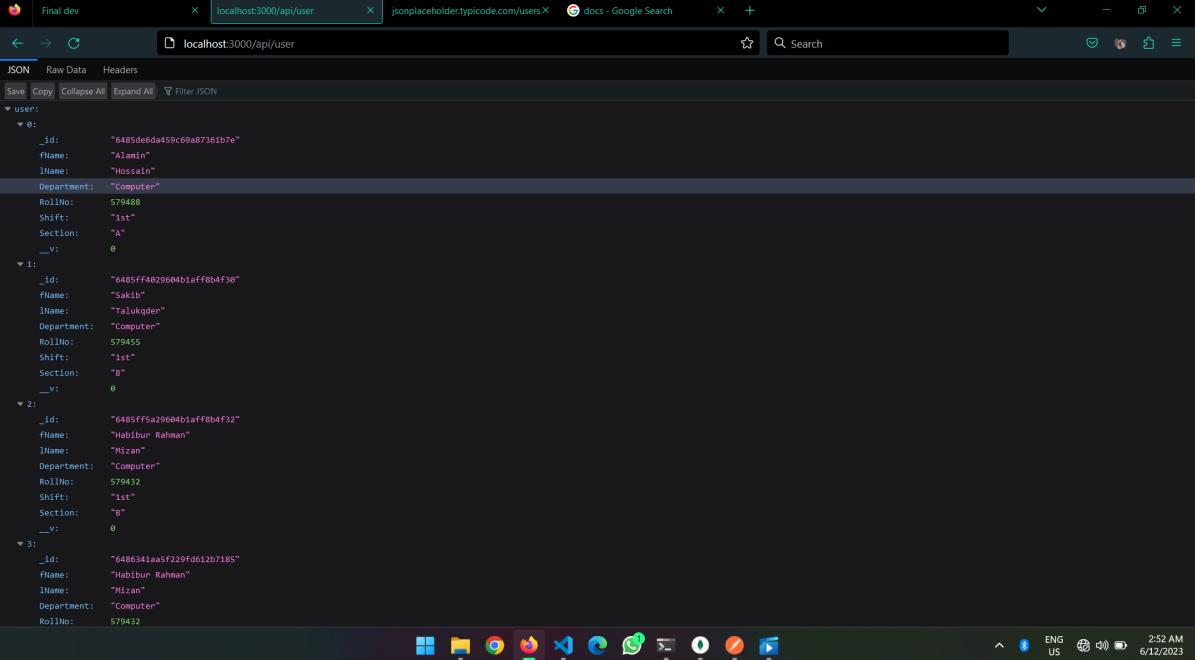


The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, Explore, and a search bar labeled "Search Postman". On the right side, there are buttons for Sign In and Create Account.

The main area displays a history of requests. A specific POST request to `http://localhost:3010/post` is selected. The request details show the method as POST, the URL as `http://localhost:3010/post`, and the body tab is active. The body content is a simple JSON object with a single key "Key": "Value".

Below the request details, the response section shows a status of 500 Internal Server Error, a time of 7 ms, and a size of 287 B. The response body contains the number 1.

Get Api:



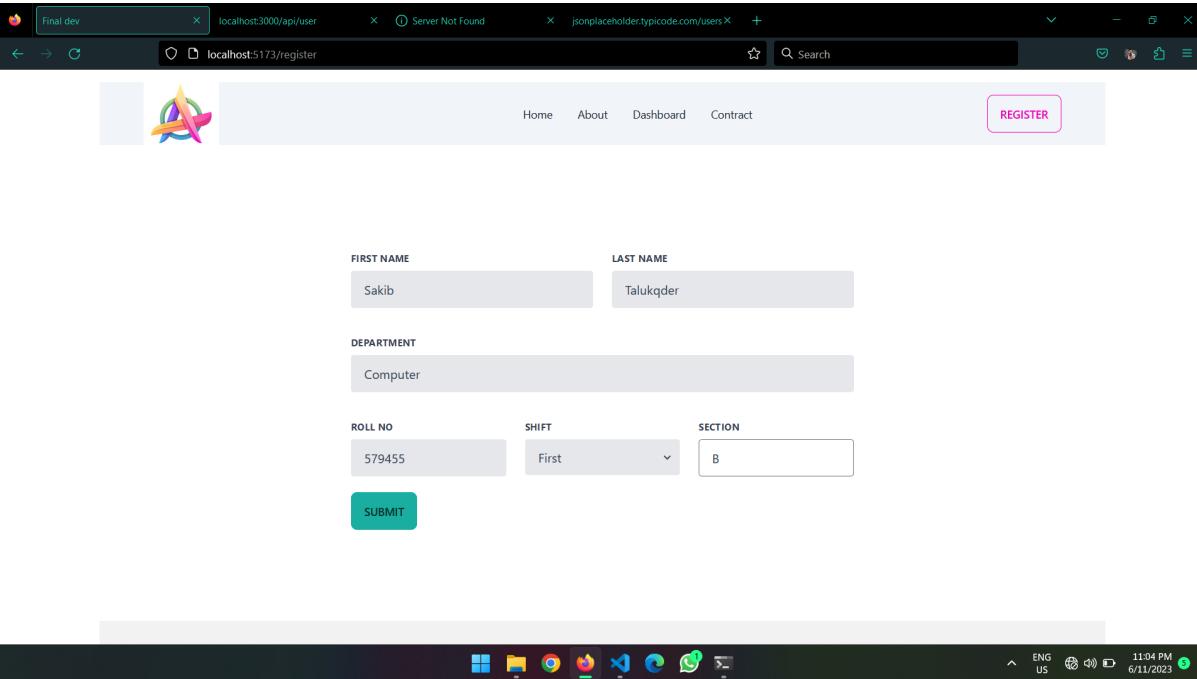
The screenshot shows a browser window with the URL `localhost:3000/api/user`. The page displays a JSON object representing multiple user records. The structure is as follows:

```

{
  "user": [
    {
      "0": {
        "_id": "6485deeda459c69a87361b7e",
        "fName": "Alamin",
        "lName": "Hossain",
        "Department": "Computer",
        "RollNo": "579488",
        "Shift": "1st",
        "Section": "A",
        "__v": 0
      },
      "1": {
        "_id": "6485ff4029604b1aff8b4f30",
        "fName": "Sakib",
        "lName": "Talukqder",
        "Department": "Computer",
        "RollNo": "579455",
        "Shift": "1st",
        "Section": "B",
        "__v": 0
      },
      "2": {
        "_id": "6485ff5a29604b1aff8b4f32",
        "fName": "Habibur Rahman",
        "lName": "Mizan",
        "Department": "Computer",
        "RollNo": "579432",
        "Shift": "1st",
        "Section": "B",
        "__v": 0
      },
      "3": {
        "_id": "6486341aa5f229fd612b7185",
        "fName": "Habibur Rahman",
        "lName": "Mizan",
        "Department": "Computer",
        "RollNo": "579432"
      }
    ]
  }
}

```

Input With Form:



The screenshot shows a browser window with the URL `localhost:5173/register`. The page contains a registration form with the following fields:

- FIRST NAME:** Sakib
- LAST NAME:** Talukqder
- DEPARTMENT:** Computer
- ROLL NO:** 579455
- SHIFT:** First
- SECTION:** B

A **REGISTER** button is located at the top right of the form area.

Mongodb Compass:

The screenshot shows the MongoDB Compass interface connected to localhost:27017. The left sidebar lists databases: admin, config, crude, dailyRoutine, dataInput, express-mongoose, local, and test. The test database is selected, and the users collection is highlighted. The main area displays the test.users collection with 1 document and 1 index. A search bar at the top is empty. Below it, tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation are visible. A query input field contains '{type a query: { field: 'value' }}'. The Documents tab shows two documents:

```
_id: ObjectId('6485ff5a29604b1aff8b4f30')
fname: "Sakib"
lname: "Talukder"
Department: "Computer"
RollNo: 579455
Shift: "1st"
Section: "B"
__v: 0

_id: ObjectId('6485ff5a29604b1aff8b4f32')
fname: "Habibur Rahman"
lname: "Mizan"
Department: "Computer"
RollNo: 579432
Shift: "1st"
Section: "B"
__v: 0
```