# 🎓 Scholarship Finder

A full-stack web application that scrapes scholarship listings from a public website and matches them to user profiles based on preferences like course, GPA, location, and interests.

## 🚀 Features

* 🔍 Web scraper built in Python to collect scholarship data.
* 🧠 Intelligent matching algorithm that uses keyword relevance and deadline urgency.
* 🧑‍🎓 User profile system to store academic and personal preferences.
* 🖥️ Responsive frontend to display matched scholarships.
* ⚡ Real-time updates with a clean, interactive UI.

## 🛠️ Tech Stack

\*\*Frontend:\*\* React

\*\*Backend:\*\* Node.js, Express

\*\*Database:\*\* MongoDB + Mongoose

\*\*Web Scraper:\*\* Python

## 📦 Folder Structure

/frontend  
 ├── App.js  
 ├── ScholarshipUI.js  
 └── index.css  
  
/backend  
 ├── controllers  
 │ ├── matchController.js  
 │ └── scrapeController.js  
 ├── models  
 │ ├── profileModel.js  
 │ └── scholarshipModel.js  
 ├── routes  
 │ ├── match.js  
 │ ├── profile.js  
 │ └── scrape.js  
 ├── scrapers  
 │ └── scrape\_scholarships.py  
 └── server.js

## ⚙️ Installation

1. 1. Clone the repository
2. 2. Install backend dependencies
3. 3. Install frontend dependencies
4. 4. Set up MongoDB
5. 5. Run the backend
6. 6. Run the frontend

## 🧪 Usage

1. Enter your profile details on the homepage (course, GPA, location, interests).  
2. Click "Find Scholarships".  
3. The system will:  
 - Save your profile.  
 - Match your profile with scraped scholarships.  
 - Display personalized results with sorting and filtering options.

## 🔄 Scraping New Scholarships

To update scholarship listings, call the following endpoint:

GET /api/scrape

## 🧩 Matching Logic

- Matches are based on:  
 - Keyword overlap in name and eligibility.  
 - Closeness of deadline (sooner = higher priority).  
- Each result is given a matchScore and sorted accordingly.

## 📦 Required Packages & Libraries

To run the project successfully, make sure to install the following dependencies:

* 🔧 Backend (Node.js)
* - express
* - mongoose
* - cors
* - body-parser
* - child\_process (built-in)
* 🖥️ Frontend (React)
* - react
* - react-dom
* - react-scripts
* 🐍 Python Scraper
* - requests
* - selenium
* - web-driver
* -pymongo

## 🙌 Acknowledgments

Made with ❤️ by ***Om G***. and ***Himanshu Kumar*** for academic and community support.