# CareerCarve 1x1 Scheduler Assignment - Reference Document

## 1. Introduction

Welcome to the CareerCarve mini-project designed for Technology Interns! The objective of this assignment is to create a 1x1 scheduler for mock interviews between MBA students and mentors. This tool will facilitate these mock interviews by ensuring both students and mentors are available at the same time for a set duration, ultimately helping students prepare for their placement processes.

### \*\*Purpose and Goals:\*\*

- Automate the scheduling of 1x1 mock interviews between students and mentors.

- Develop a user-friendly interface for both students and mentors to manage their schedules.

- Ensure that the scheduling respects student preferences and mentor availabilities.

- Facilitate payments for premium services such as selecting a preferred mentor.

## 2. Step-by-Step Instructions

### \*\*Step 1: Setting Up the Development Environment\*\*

1. **Install Node.js**: Ensure you have the latest version of Node.js installed.

2. **Set Up MySQL**: Install MySQL and configure it for use in your development environment.

3. **Install React or React Native**: Depending on whether you are developing for the web or a mobile platform, set up React (for web) or React Native (for mobile).

4. **Optional - Additional Tools**: Install other tools or libraries you might need, such as Visual Studio Code, Postman, etc.

### \*\*Step 2: Planning the Project\*\*

1. **Understand the Requirements**: Review the assignment description and make sure you understand each requirement fully.

2. **Design the UI**: Plan out the user interface for both students and mentors. Consider sketching wireframes or using a design tool like Figma.

3. **Database Schema**: Draw a schema for your database, focusing on tables like `students`, `mentors`, `schedules`, and `payments`.

### \*\*Step 3: Develop the Backend API (if you're handling backend)\*\*

1. **Initialize Node.js Project**: Create a new Node.js project and install necessary packages (Express, MySQL, etc.).

2. **Database Connection**: Set up the connection to your MySQL database.

3. **Build RESTful APIs**: Create APIs for functionalities including:

- Fetching student and mentor availability.

- Creating and updating schedules.

- Processing payments.

4. **Implement Business Logic**: Write the logic to allocate schedules, prioritize mentor availability, and handle premium services.

5. **Testing**: Write unit tests and integration tests for your APIs.

### \*\*Step 4: Develop the Frontend (if you're handling frontend)\*\*

1. **Initialize React Project**: Create a new React project (or React Native for mobile).

2. **Build User Interfaces**: Develop the UI components based on your wireframes. Key screens to build:

- **Login/Signup Screen**: For both students and mentors.

- **Dashboard**: To view and manage schedules.

- **Scheduling Form**: For students to book mock interviews.

- **Payment Page**: To handle premium service payments.

3. **API Integration**: Connect your frontend with the backend APIs.

4. **Handle User Flows**: Ensure the user flow is seamless and intuitive, covering scenarios like selecting a mentor, choosing a time slot, and completing payments.

5. **Testing**: Conduct thorough testing to ensure the UI is responsive and user-friendly.

### \*\*Step 5: Finalizing and Submitting the Assignment\*\*

1. **Review Your Work**: Check each part of your project for completeness and quality.

2. **Prepare for Submission**: Ensure all your code is documented, and test cases are included.

3. **Live Demonstration**: Prepare a brief presentation to showcase your scheduler and explain your approach.

## 3. Best Practices

### \*\*Code Quality\*\*

- Write clean and modular code.

- Follow standard coding conventions and best practices.

### \*\*User Experience\*\*

- Focus on creating an intuitive and responsive design.

- Minimize user input errors by providing clear instructions and validation messages.

### \*\*Testing\*\*

- Cover your code with unit and integration tests.

- Perform user acceptance testing to ensure the tool meets initial requirements.

### \*\*Efficiency\*\*

- Optimize database queries and API performance.

- Aim to reduce load times and improve overall speed.

### \*\*Documentation\*\*

- Document your codebase clearly.

- Provide a README file with setup instructions and any necessary information.

## 4. Submission Guidelines

1. **Code Repository**: Push your code to a Git repository (GitHub, GitLab, etc.) and share the repository link.

2. **Screenshots/Videos**: Include screenshots or a video demonstration showing the main features and flows of your scheduler.

3. **Documentation**: Ensure all your documentation is up-to-date and clear.

4. **Submit via Email**: Send your repository link and any necessary details to swaroop@careercarve.com before the deadline.

## 5. Frequently Asked Questions (FAQ)

### 1. \*\*What should I do if I don't know Node.js or MySQL very well?\*\*

- Focus on the part of the assignment you're comfortable with. If you handle the frontend, store data in JSON format for simplicity.

### 2. \*\*How should I prioritize tasks given the limited time frame?\*\*

- Start with the key functionalities: scheduling logic and UI for booking. Then, move on to additional features like payment.

### 3. \*\*What if I encounter blockers during development?\*\*

- Look for resources online, or ask for help via email (swaroop@careercarve.com).

### 4. \*\*How can I test my scheduler effectively?\*\*

- Test different scenarios for scheduling, such as different durations, mentor preferences, and busy schedules. Also, ensure payment flows are working correctly.

### 5. \*\*Can I use additional libraries or frameworks not mentioned in the assignment?\*\*

- Yes, as long as they help you achieve the task more efficiently and effectively.

Good luck with your assignment, and we look forward to seeing your innovative solution!