



**DEAR
FRESHER**



কনফিউজং জব মার্কুলার

Sr. Software Developer (Full-Stack Developer, Docker, Laravel, React.js/Vue Js, Python)

[View all jobs of this company](#)

Job Context

- We are looking for a expert React js Front-end, Back-end, Docker, Laravel, React.js/Vue Js, Python, developer. Primary focus of this position will be on developing user interface components and implementing them following well-known React js workflows. Need to ensure that these components and the overall application are robust and easy to maintain. Need to coordinate with the rest of the team working on different layers of the infrastructure. Therefore, a commitment to collaborative problem solving, sophisticated design, and quality product is important.
- Our innovative and talented team members from all around the world are working hard in the areas of E-commerce, Web & Graphic designs, Software & Mobile apps development, Web hosting, Motion graphics and 3D modeling, Search Engine Optimization (SEO), Digital marketing, Information technology and bringing outstanding success for the entire company.

Job Responsibilities

- Write complex backend code with **Node.js** (preferred)/PHP with Laravel.
- Write well designed, testable, efficient code.
- Clear understanding of OOP, Data Structures, Algorithms and Design Patterns.
- Create automated builds and test with Docker is a plus.
- Independently plan, design, maintain and deploy web applications to meet business needs.
- Research, guide and solve complex performance problems and architectural challenges.
- Develop and deploy new features to facilitate related procedures and tools if necessary.
- Ensure the accurate operation of web servers and software.
- Respond promptly and professionally to bug reports.
- Support continuous improvement by investigating alternatives and technologies and presenting these for architectural review.
- Stay up-to-date with new technologies and continuously evaluate and promote the use of new tools, techniques, frameworks, and platforms.
- Provide technical guidance and assistance to other Jr. Software Engineers.
- Building reusable components and front-end libraries for future use.

হবি প্রতীকী

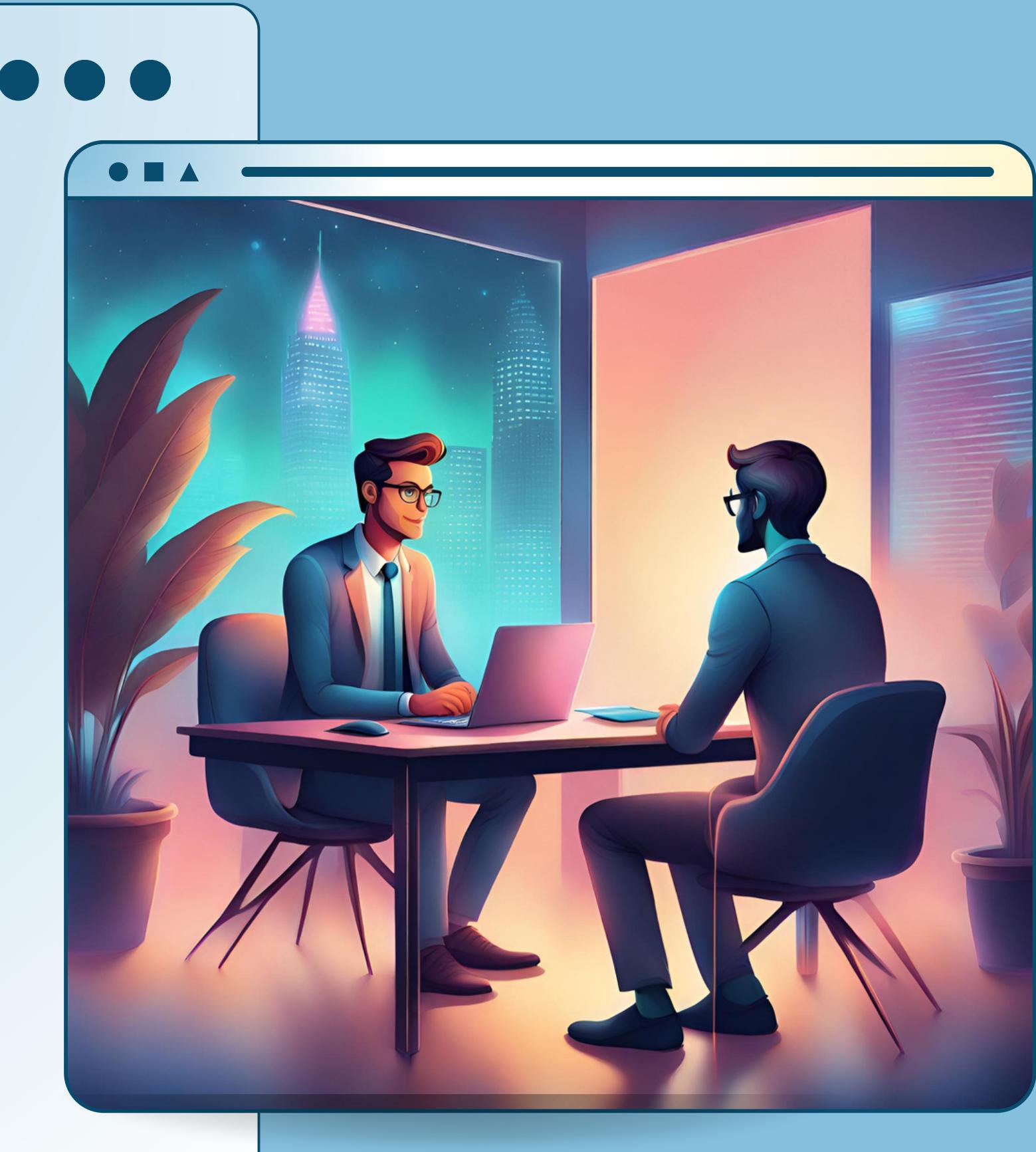
- Must be a full stack PHP, Laravel Developer (Front-end and Back-end)
- Must have experience in teamwork and lead a team in a previous job
- At least 5 year(s) experience with PHP and Laravel
- Good understanding of Laravel Framework and modern PHP (PHP7/PHP8)
- Good experience with Vue.JS, Nuxt.JS, Rest API
- Excellent skills in modern JavaScript (ES6+)
- Good working knowledge of Git, GitHub, Jira, and good security practices
- Responsive design using CSS and HTML5, Bootstrap 4/5
- Strong knowledge of web services in Apache and Nginx.
- Experienced in service integrations (Web Services, CMS/E-commerce systems, 3rd party APIs, etc.)
- Experience with both consuming and designing RESTful APIs.
- Must have Experience in maintaining Linux server, VPS (expert in using terminal)
- Must have skills in leadership, team management, and problem-solving capabilities

Employment Status

Full-time

THE REALITY

- সিমিলার স্কিল সেট বোর্ডানো ।
- কয়েকটি ভ্যাকেন্সির জন্য একটি জব পোষ্ট ।
- কোম্পানির একাধিক টার্গেট থাকা ।
- ডিন স্কিল সেট থেকে বেছ্টে বাছাই করা ।
- অনেক সময় না বুঝে লেখা ।



HOW FRESHER CHOOSE RIGHT JOB POST ?

- କିମ୍ବା କିମ୍ବା ଶିଳ୍ପ ମୂଲ୍ୟ କରେ କି ନା ?
- ଅଭିଜ୍ଞତା ୦- ୧/୨/୩ ବର୍ଷରେ ମଧ୍ୟେ ଆଛେ କି ନା ?
- ସମ୍ପଦ କତ ଚାହେଚେ ?
- ରେସ୍ପନ୍ସିବିଲିଟି ୩୦-୮୦% ମୂଲ୍ୟ କରେ କି ନା?



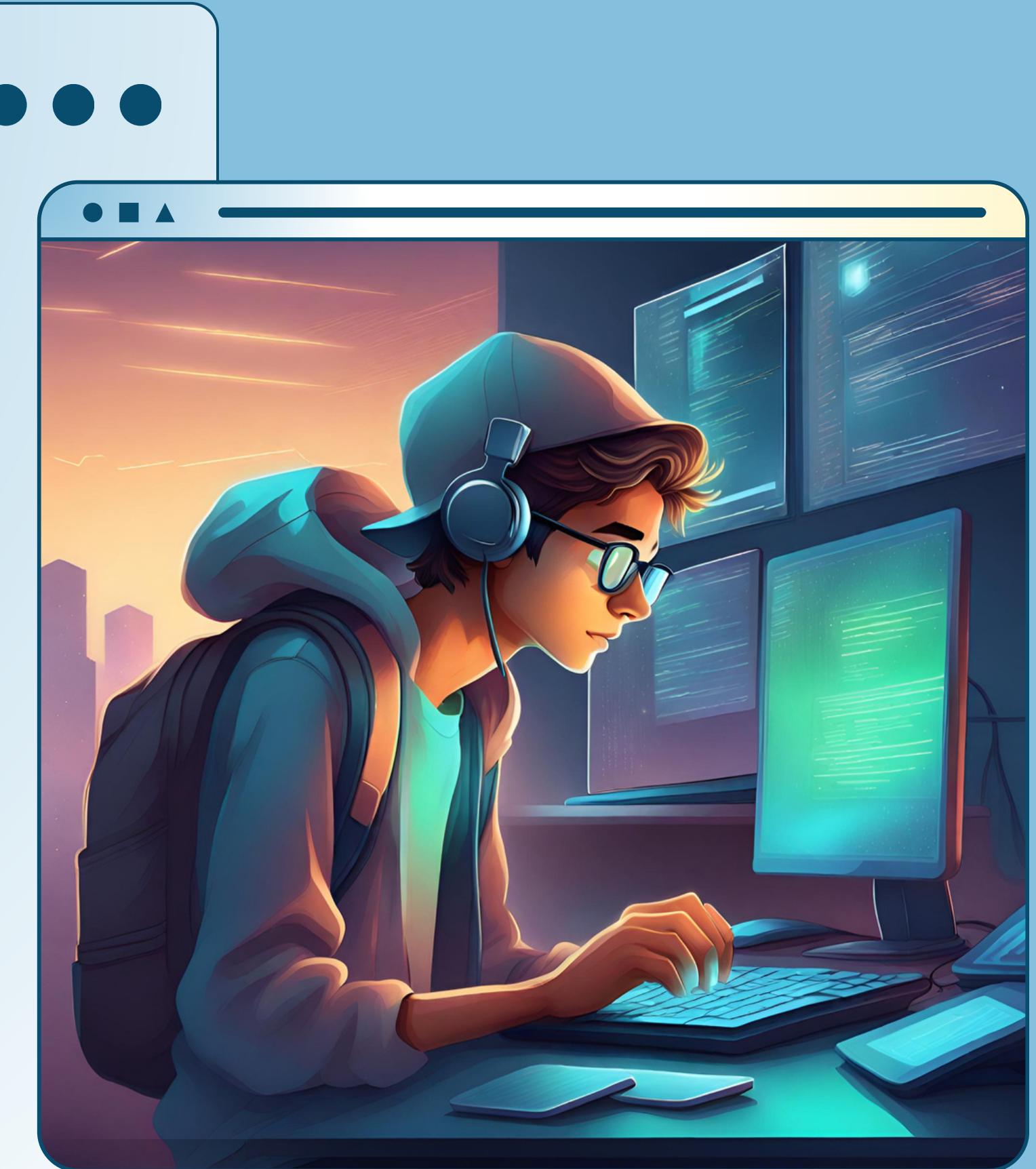
ফ্রেশার হলেও আপনি ডাক পাবেন যদি -

- আপনার সিভি-স্কিল-পোর্টফোলিও ম্যাচ করে
- সুন্দর ডেভেলপার প্রোফাইল থাকে
- সোশাল মিডিয়া প্রোফাইল মাস্ট



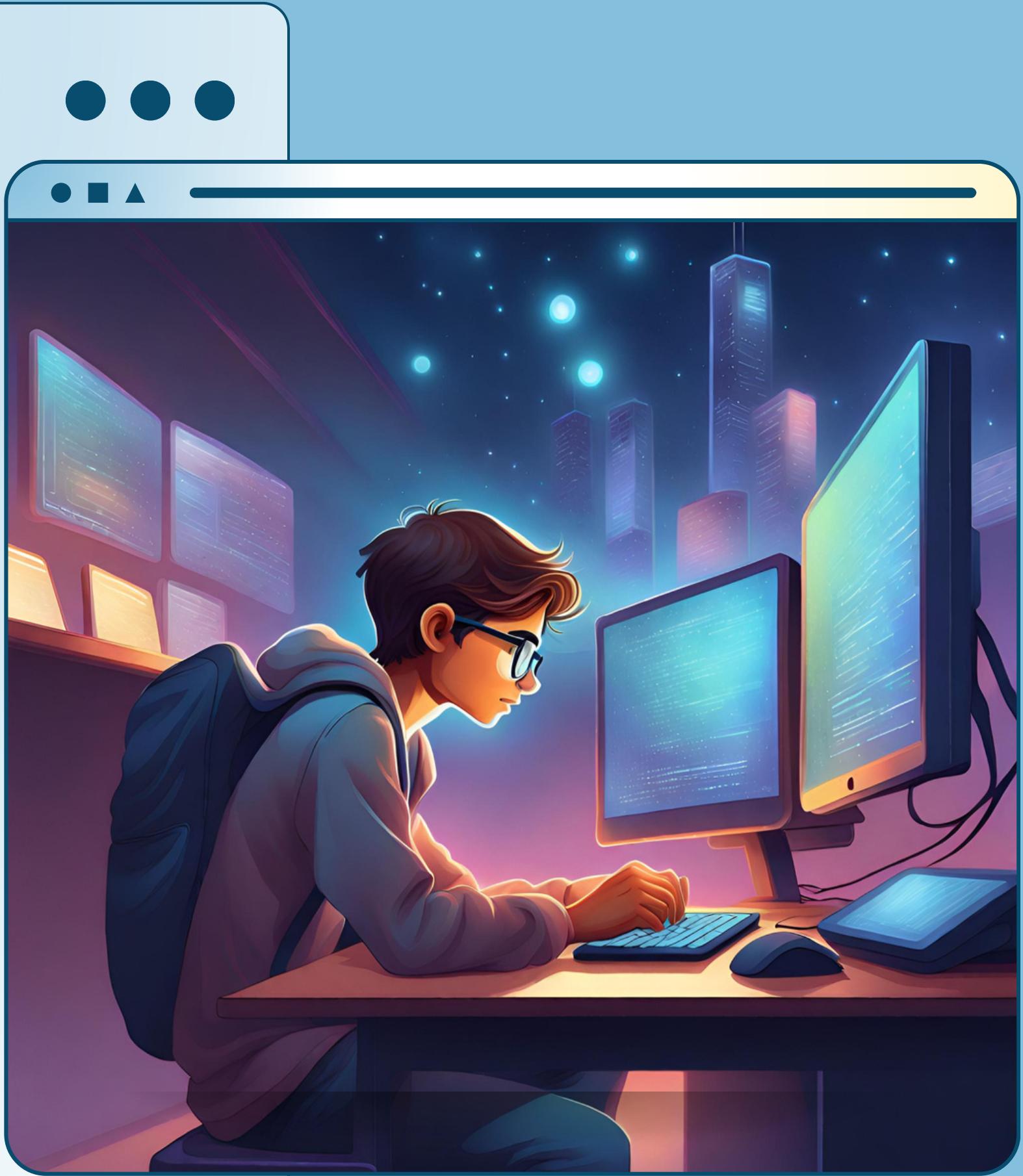
ରିମୋଟ ଜବ କୋଥାଯି ମାର୍ଚ କରବେନ ?

- LinkedIn
- Working Nomads
- SkipTheDrive
- Upwork
- Turing
- FlexJobs
- Virtual Vocations
- Remote.co
- Indeed
- We Work Remotely



লোকাল জব কোথায় সার্চ করবেন ?

- **LinkedIn**
- **BDJOBS**
- **Social Media Group/Page**



কোম্পানি সম্পর্ক জানা জারি

- Company Background
- Products and Services
- Company Culture
- Leadership
- Industry Position
- Financial Health
- Clients and Partnerships
- Press and Media
- Work and Projects

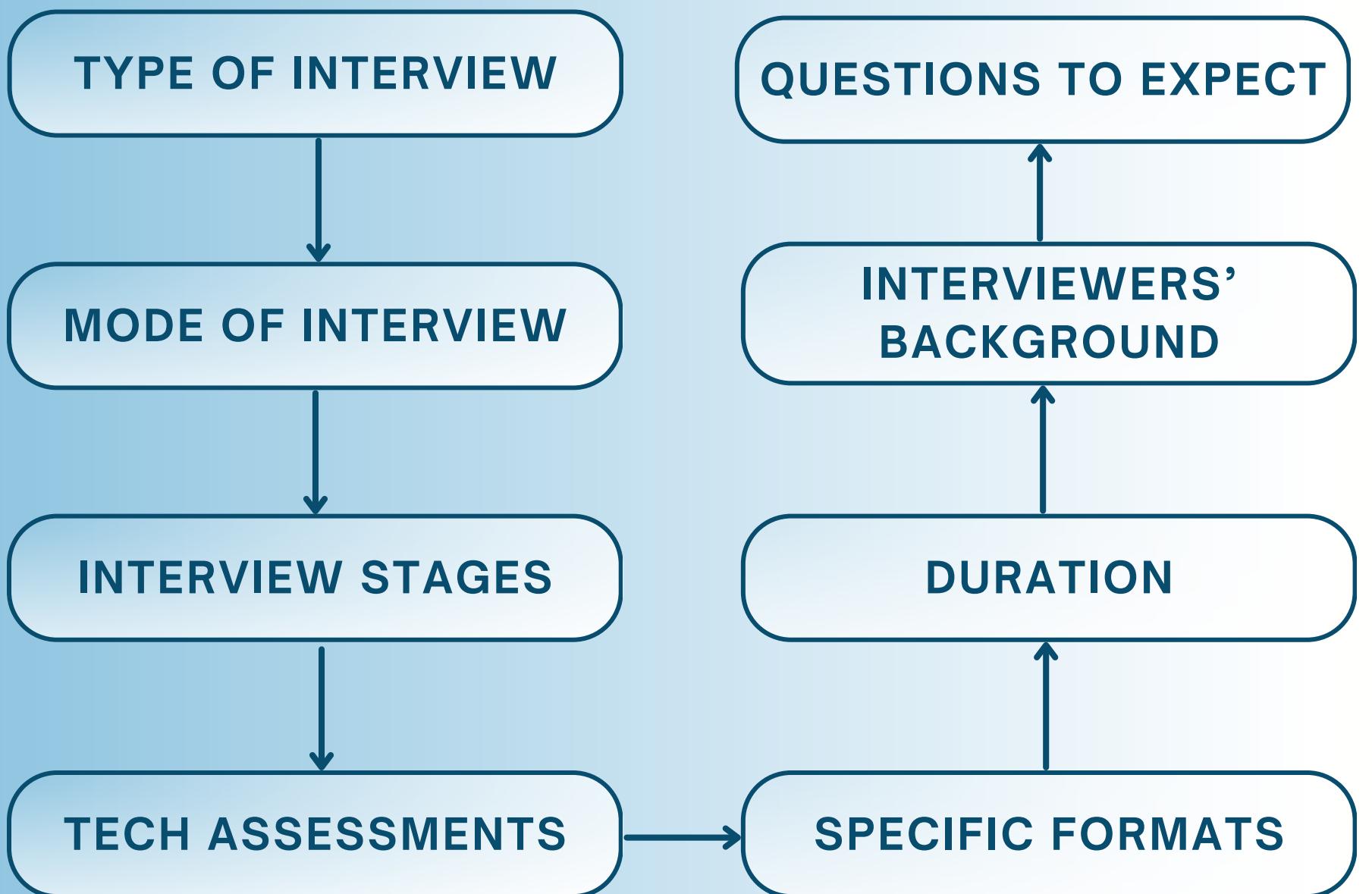


সফট স্কিলের বিকল্প নাই

- Communication
- Teamwork
- Work Ethic
- Problem-Solving
- Leadership
- Adaptability
- Time Management
- Emotional Intelligence
- Critical Thinking
- Self-Improvement
- Interpersonal Skills
- Cultural Competence
- Personal Attributes
- Accountability

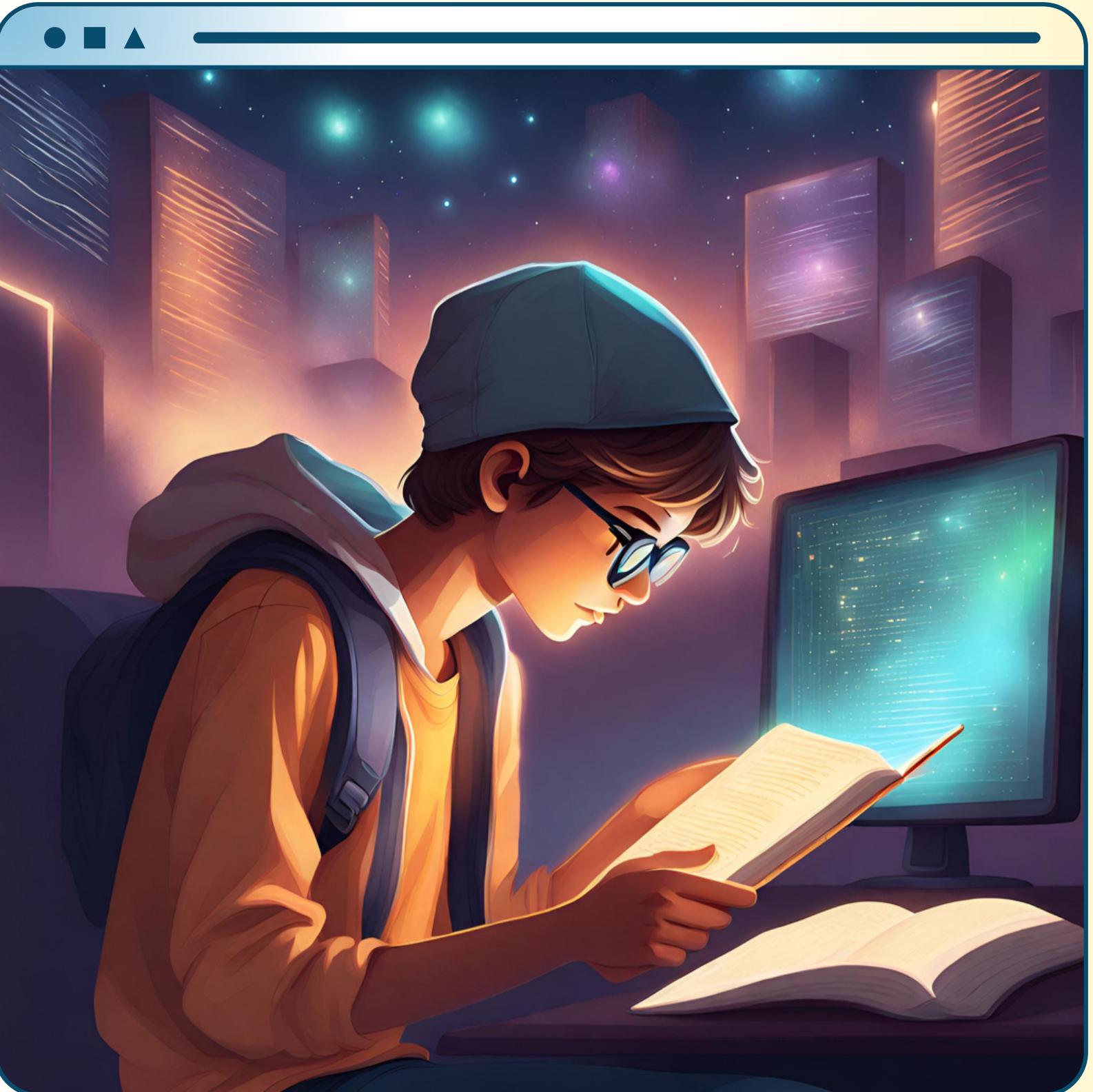


ইন্টারভিউ প্রসেস না জানলে আপনি শুরুতেই ফেঁস !



ফ্রেশারদের যে বৈশিষ্ট্য গুলো ওকত্ত পায়

- সেলফ এন্ড কুইক লার্নিং ক্যাপাবিলিটি
- কোম্পানিতে ভেলু এড করার লক্ষ্য-উদ্দেশ্য
- টিম-ওয়ার্ক করার ক্যাপাবিলিটি
- পড়াশুনা-হার্ড ওয়ার্ক ক্যাপাবিলিটি
- ক্যারিয়ার গোল এন্ড ডিশন



ফ্রেশার বেতন কত চাইবে?

- শুরুতেই কোনো বার্গিং নয়
- AS PER COMPANY POLICY



JavaScript Basics

- Data types (strings, numbers, boolean, null, undefined)
- Variables (var, let, const) and scoping rules
- Operators (assignment, comparison, arithmetic, logical)
- Control structures (if-else, switch, loops)

JavaScript Functions

- Declaration vs. expressions
- Arrow functions
- Callbacks
- Scope and closures
- Function hoisting



JavaScript

JavaScript DOM

- Selecting and modifying elements
- Event handling
- Creating and appending elements
- Document and window objects

ES6+ Features

- Template literals
- Destructuring
- Default parameters
- Spread/rest operators
- Array and object methods



JavaScript

Asynchronous JavaScript

- Event loop and the concept of non-blocking
- Promises, async/await
- Making HTTP requests (XMLHttpRequest, fetch API)

Error Handling

- try/catch blocks
- Throwing errors
- Error propagation and custom error objects



JavaScript

Modules

- Import/export syntax
- Module resolution
- Bundlers (like Webpack)

Testing

- Basic concepts of unit testing
- Familiarity with testing libraries (e.g., Jest)

Development Tools

- Debugging using browser developer tools
- Version control with Git
- Basic command-line usage



JavaScript

Code Quality

- Coding standards and best practices
- Code readability and maintainability
- Refactoring techniques

Basic Security Concepts

- Cross-site scripting (XSS)
- Cross-origin resource sharing (CORS)

Understanding the Web

- Basic knowledge of how the internet works
- HTTP/HTTPS and web protocols
- RESTful services and APIs



JavaScript

React Basics

- Understanding JSX and its syntax.
- Components and Props: Functional vs. Class components, and passing props.
- State Management: useState hook, class component state, and state lifting.

React Lifecycle:

- Lifecycle methods for class components.
- useEffect hook for functional components.



REACT

Hooks

- Core hooks: `useState`, `useEffect`, `useContext`.
- Additional hooks: `useReducer`, `useCallback`, `useMemo`, `useRef`, and custom hooks.

Event Handling

- Handling events in JSX (`onClick`, `onSubmit`, etc.).
- Binding 'this' in event handlers.

Conditional Rendering

- Techniques for rendering components conditionally.
- Inline conditional expressions and logical operators.



REACT

Lists and Keys

- Rendering lists using the `map()` method.
- Importance of keys for list items and their correct usage.

Form Handling

- Controlled components.
- Handling form submissions.

Component Composition and Reusability

- Higher-order components (HOCs).
- Render props and component composition.



REACT

Context API

- Using Context to pass data through the component tree.
- When to use Context vs. props.

Routing with React Router

- Basic routing concepts.
- Route parameters and navigation.

React and HTTP

- Making HTTP requests in React using fetch or Axios.
- Handling responses and state updates based on API calls.

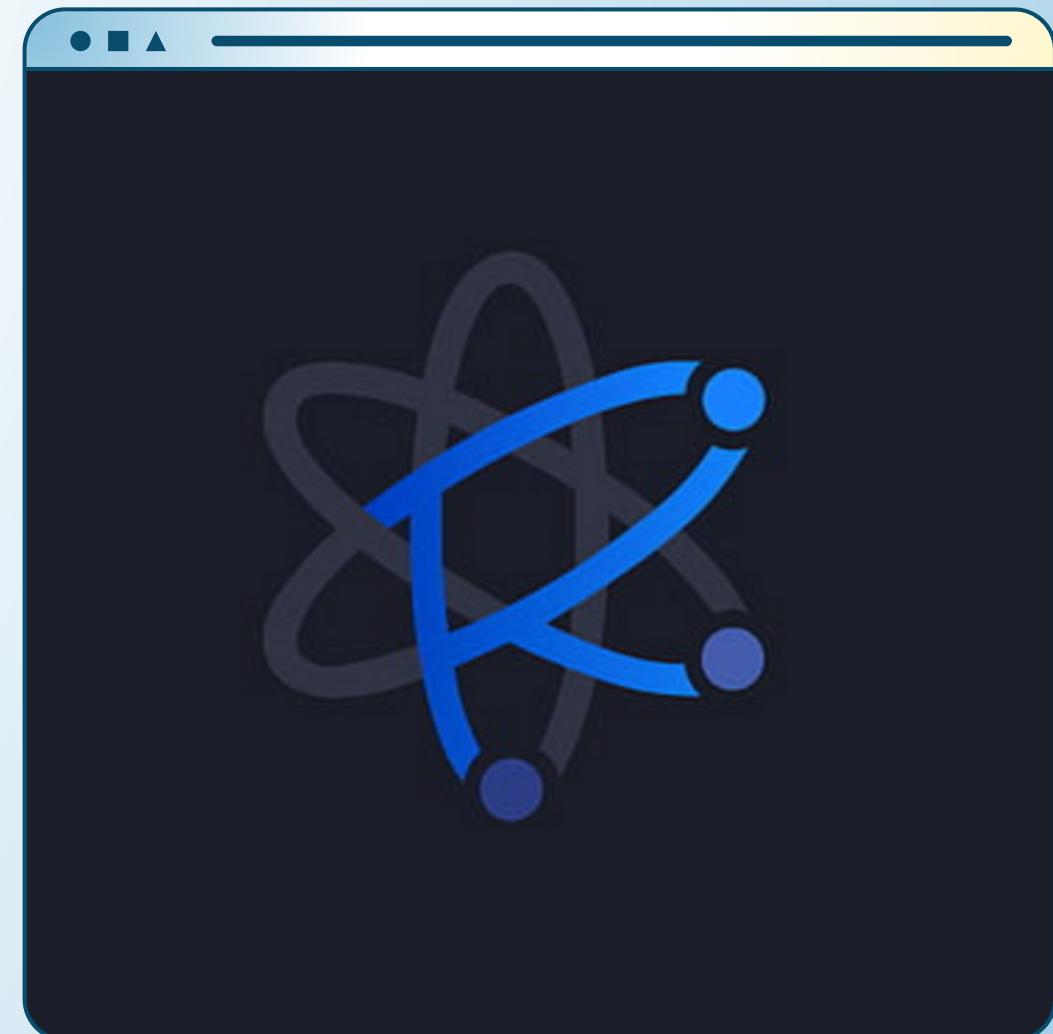


React Ecosystem

- Overview of common libraries and tools used in React projects (e.g., Formik for forms, React Query for data fetching).

State Management Libraries

- Overview of Redux or Context API for state management.
- Actions, reducers, and store concepts in Redux.



REACT

Node.js Basics

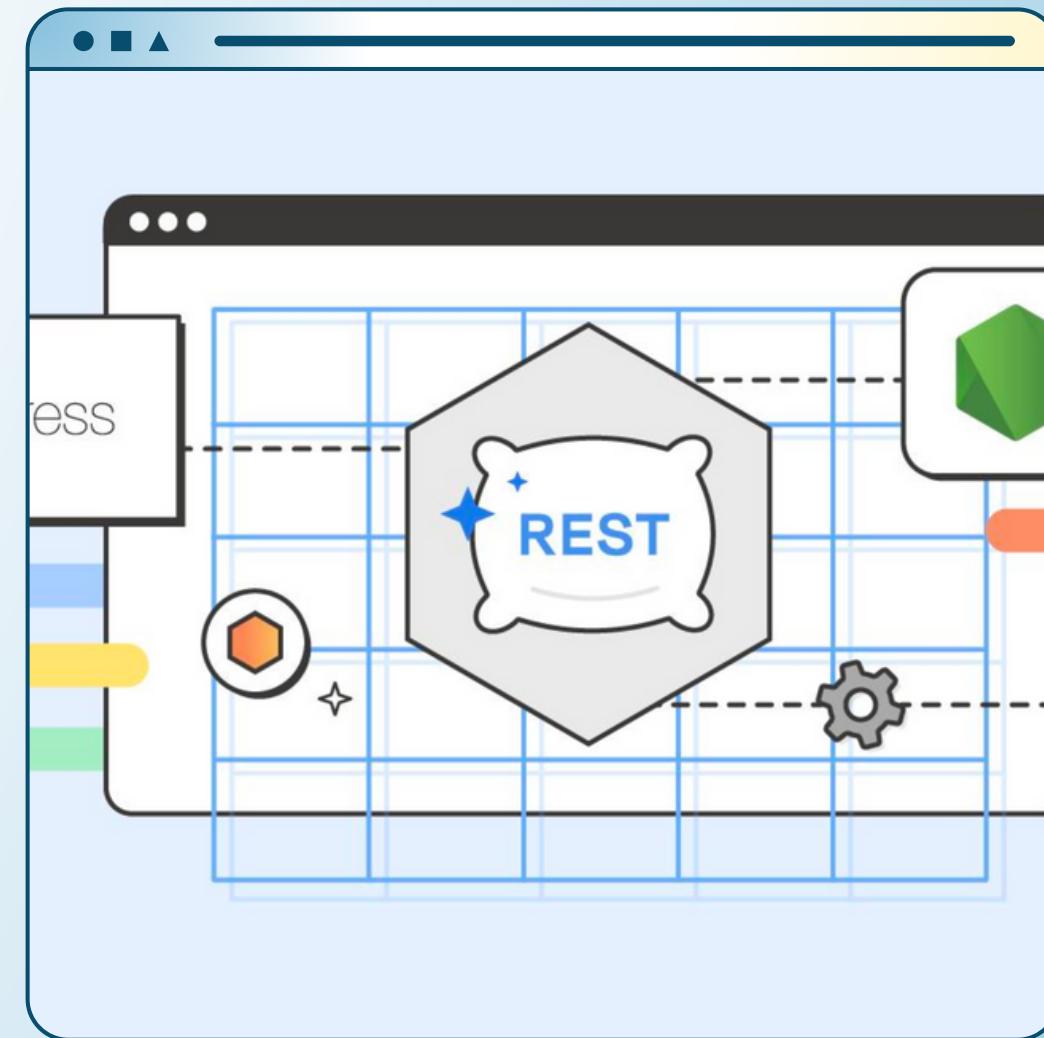
- Understanding of Node.js and its non-blocking nature.
- Core modules such as HTTP, file system, path, and events.

Express.js Fundamentals

- Setting up an Express server.
- Understanding middleware and its usage.
- Handling routing with Express.

HTTP Methods and RESTful Services

- Knowledge of HTTP verbs (GET, POST, PUT, DELETE).
- Creating RESTful APIs with Express.



EXPRESS JS

Request and Response Objects

- Working with `req` (request) and `res` (response) objects.
- Handling query parameters, body, and URL parameters.

Middleware

- The role of middleware in Express.
- Using built-in middleware like `express.json()` and `express.urlencoded()`.
- Implementing third-party middleware (e.g., `body-parser`, `morgan`, `cors`).



EXPRESS JS

Error Handling

- Writing error-handling middleware.
- Catching and responding to errors.

Database Integration

- Connecting an Express app to a database (e.g., MongoDB, PostgreSQL).
- Using ORMs like Sequelize or Mongoose.

Authentication

- Basics of implementing authentication
- Understanding sessions and JWT for stateless authentication.



EXPRESS JS

Environment Variables

- Using environment variables for configuration

API Testing

- Writing tests for API endpoints.
- Postman Documentation

Package Management

- Using npm or Yarn for package management.
- Understanding package.json and managing dependencies.



EXPRESS JS

Security Practices

- Knowledge of basic security practices, such as input validation, to prevent SQL injection, XSS attacks, etc.

Deployment

- Basics of deploying an Express application

Logging Debugging

- Implementing logging Debugging in an Express app

Code Quality

- Following coding standards and best practices.
- Writing maintainable and readable code.



EXPRESS JS

NoSQL vs. SQL

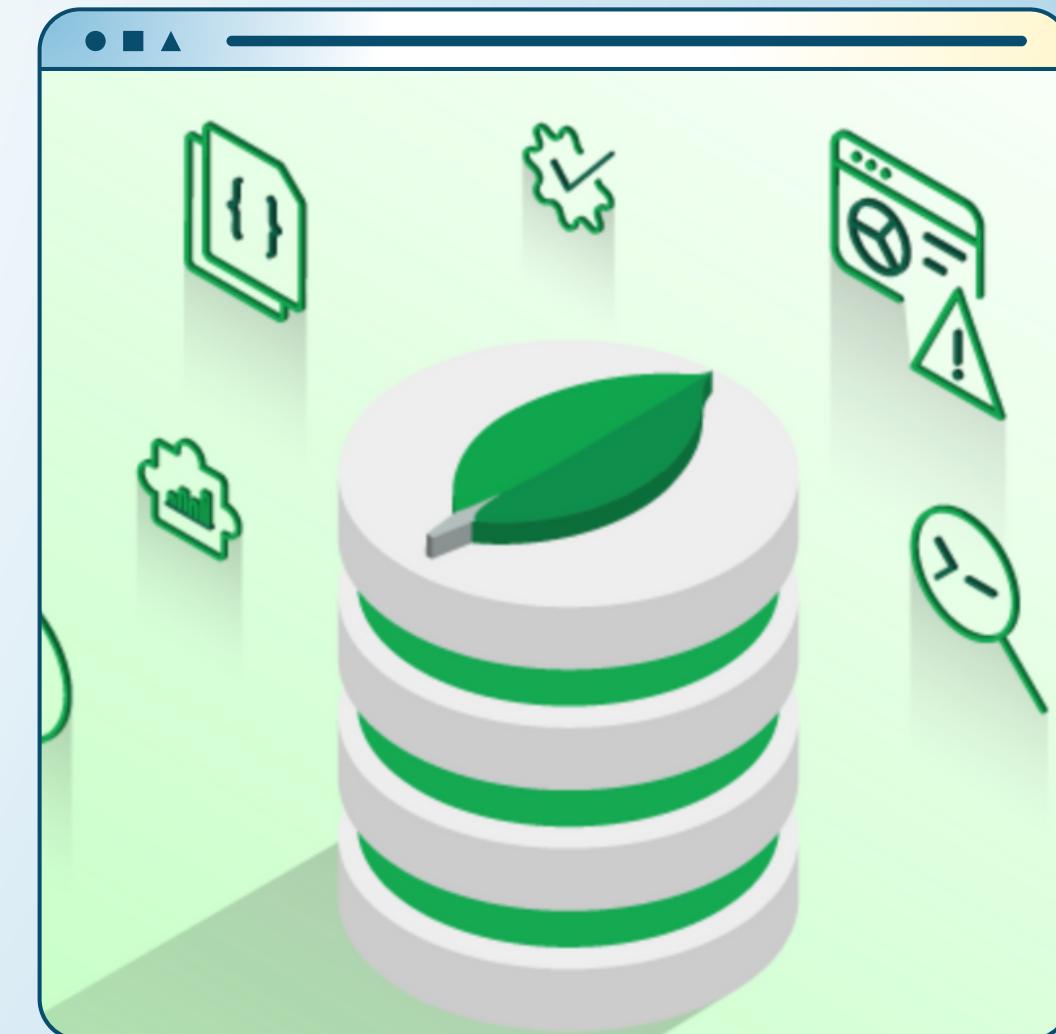
- Understanding the differences between NoSQL databases like MongoDB and traditional SQL databases.
- When and why to use NoSQL databases.

MongoDB Basics

- Installation and setup of MongoDB.
- MongoDB shell commands.
- Understanding MongoDB's document-oriented structure.

CRUD Operations

- Creating, reading, updating, and deleting documents.
- Writing MongoDB queries to perform CRUD operations.



Data Modeling

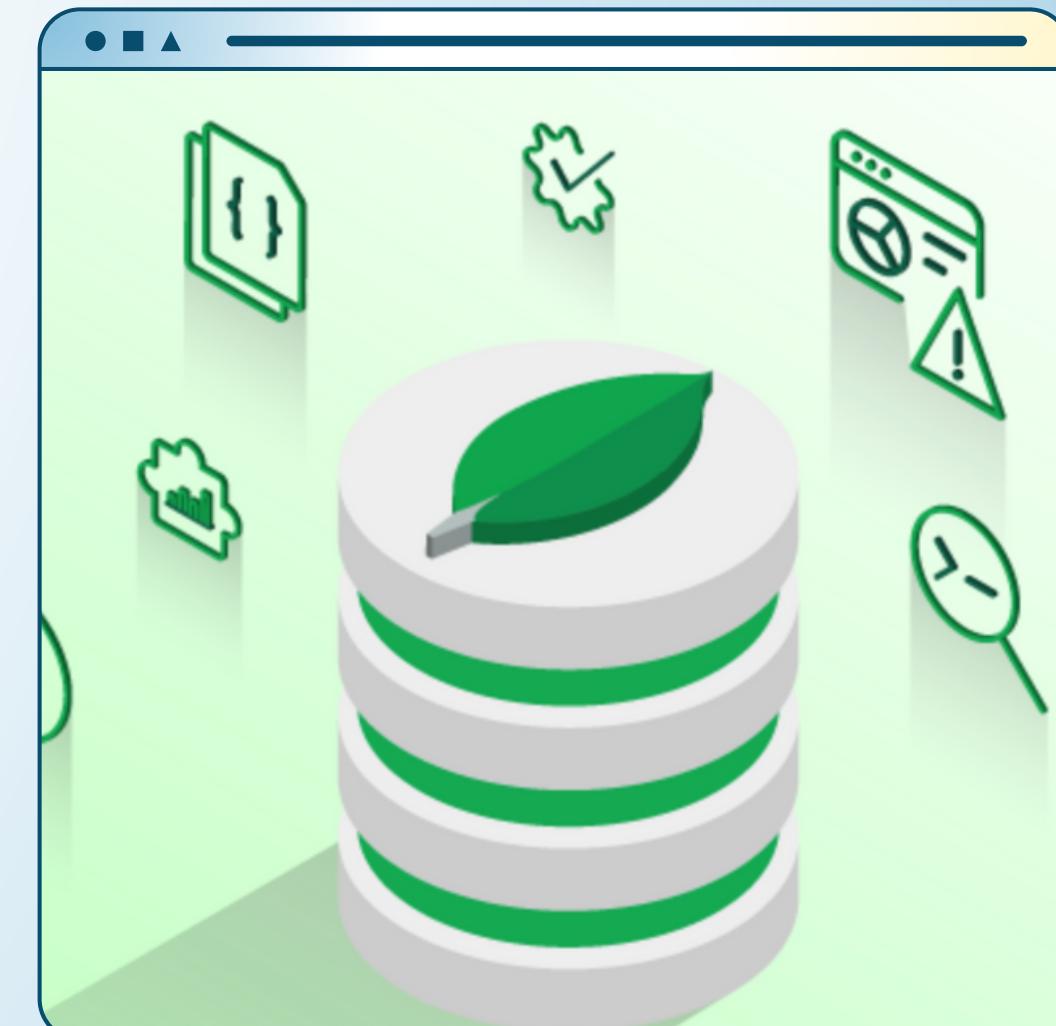
- Understanding BSON data format.
- Designing schemas using MongoDB's flexible schema approach.

Indexes

- Purpose of indexing in MongoDB.
- Creating and managing indexes.

Aggregation Framework

- Using the aggregation pipeline to transform and combine data.
- Common aggregation operators and stages.



Mongo DB

Transactions

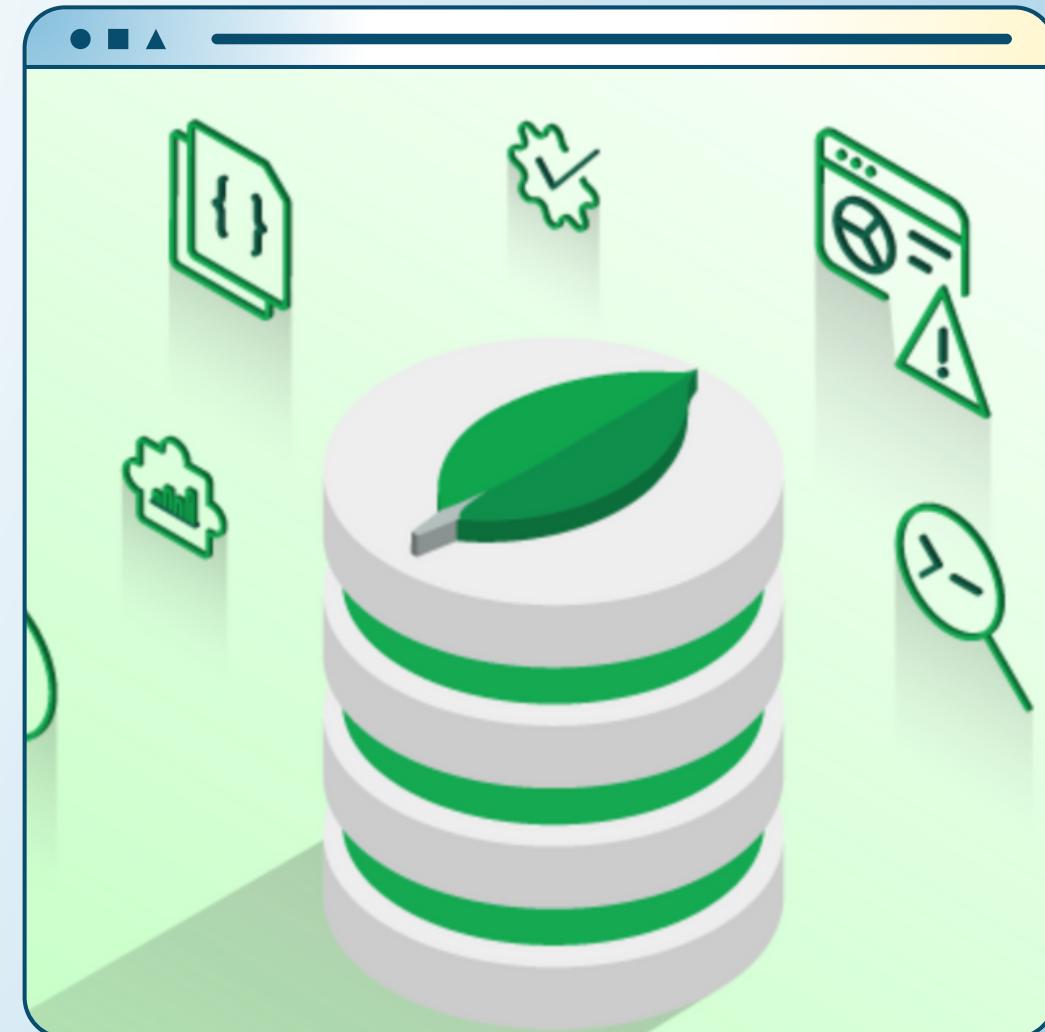
- How to handle transactions in MongoDB.
- Limitations and best practices for using transactions.

MongoDB Atlas

- Using MongoDB's cloud service, Atlas.
- Setting up clusters and managing databases in the cloud.

MongoDB Tools

- Familiarity with tools like MongoDB Compass



Mongo DB

Related Practice Projects

