JAVASCRIPT Junior

**Загальні:**

1.Які методи HTTP-запитів ви знаєте?

2.Які версії HTTP-протоколу вам відомі?  
3.Які знаєте коди відповіді (стану) HTTP?  
4.Що таке Cross-Origin Resource Sharing? Як усунути проблеми з CORS?  
5.Що таке cookie?  
6.Який максимальний розмір cookie?  
7.Що означає директива use strict?  
8.Чим JS відрізняється під час роботи на front-end і back-end?  
9.Що таке статична і динамічна типізації?  
10.Як клієнт взаємодіє із сервером?  
11.Що таке REST?  
12.Поясніть поняття мутабельність / іммутабельність? Які типи є мутабельними й навпаки?  
13.Як шукати помилки в коді? Чи використовуєте дебагер?  
14.Яких відомих людей зі світу JS знаєте?

**JS Core**

15.Які існують типи даних у JS?  
16.Як перевірити, чи об’єкт є масивом?  
17.Як перевірити, чи число є скінченним?  
18.Як перевірити, що змінна рівна NaN?  
19.Чим відрізняється поведінка isNaN() та Number.isNaN()?  
20.Порівняйте ключові слова var, let, const.  
21.Що таке область видимості?  
22.Що таке деструктуризація?  
23.Для чого призначені методи setTimeout і setInterval?  
24.Порівняйте підходи роботи з асинхронним кодом: сallbacks vs promises vs async/await.  
25.Чи можна записувати нові властивості / функції в прототипи стандартних класів (Array, Object тощо)? Чому ні? У яких випадках це робити можна? Як убезпечити себе, якщо потрібно розширити прототип?  
26.Назвіть методи масивів, які пам’ятаєте, і скажіть, для чого вони потрібні.  
27.Які перебираючі методи масиву знаєте? У чому їхня відмінність?  
28.Як працюють оператори присвоєння / порівняння / рядкові / арифметичні / бітові тощо?  
29.Опишіть призначення і принципи роботи з колекціями Map і Set.  
30.Що означає глибока (deep) та поверхнева (shallow) копія об’єкта? Як зробити кожну з них?

**Функції:**

31.Яка різниця між декларацією функції (function declaration) та функціональним виразом (function expression)?  
32.Що таке анонімна функція?  
33.Розкажіть про стрілкові функції (arrow function). В чому полягають відмінності стрілкових функцій від звичайних?  
34.Що таке і для чого використовують IIFE (Immediately Invoked Function Expression)?  
35.Що таке hoisting, як він працює для змінних і функцій?  
36.Що таке замикання (closure) і які сценарії його використання?  
37.Як ви розумієте замикання? Що буде виведено в консолі в цьому випадку?

var f = function() {

  console.log(1);

}

var execute = function(f) {

  setTimeout(f, 1000);

}

execute(f); *// что выведет в консоль и почему*

f = function() {

  console.log(2);

}

38.Що таке рекурсія?  
39.Що означає ключове слово this?  
40.Що таке втрата контексту, коли відбувається і як їй запобігти?  
41.Методи функцій bind / call / apply — навіщо і в чому різниця?

**Front-end**

42.Що таке DOM?  
43.Порівняйте атрибути підключення скрипту async і defer в HTML-документі.  
44.Яка різниця між властивостями HTML-елементів innerHTML і innerText?  
45.Опишіть процес спливання (bubbling) подій у DOM.  
46.Як зупинити спливання (bubbling) події?  
47.Як зупинити дефолтну обробку події?  
48.Чому дорівнює this в обробнику подій (event handler)?  
49.Що таке LocalStorage і SessionStorage? Який максимальний розмір LocalStorage?  
50.Як отримати висоту блоку? Його положення щодо меж документа?  
51.Що таке webpack?  
52.Чим відрізняється dev-збірка від prod?

**Верстка**

53.Що таке блокова модель CSS?  
54.Які способи центрування блокового контенту по горизонталі та вертикалі знаєте?  
55.Які підходи у верстці вам відомі (float, flex, grid, etc)?  
56.Як зробити додаток responsive?  
57.Які є принципи семантичної верстки?  
58.Навіщо потрібні префікси для деяких CSS-властивостей (-webkit-, -moz- тощо)?  
59.Як спростити написання кросбраузерних стилів?  
60.Практичне завдання: прокоментувати та виправити приклад поганого CSS або HTML.  
61.Що таке CSS-препроцесори? З якими працювали? Що нового вони приносять у стандартний CSS?

**Angular**

62.Перерахуйте основні компоненти фреймворку (модуль, роут, директива тощо).  
63.У чому різниця між компонентом і директивою?  
64.Розкажіть про життєвий цикл компонента.  
65.Перерахуйте часто використовувані хуки життєвого циклу компонента та розкажіть, для чого вони потрібні?  
66.У чому різниця між конструктором і ngOnInit-хуком?  
67.Як захистити роут від несанкціонованого доступу? Які механізми надає для цього фреймворк?  
68.Що таке Lazy loading, як і для чого використовується?  
69.Яке призначення RouterOutlet?  
70.Як компоненти можуть взаємодіяти один з одним?  
71.Як створити two-way binding властивість для компонента?  
72.Які типи форм є у фреймворку? У яких випадках і що краще використовувати?  
73.Які стани є у форми і як це можна застосувати?  
74.Навіщо потрібні сервіси? Як з ними працювати?  
75.Що таке singleton-сервіси? Яке їхнє призначення? Спосіб створення?  
76.Які є способи оголошення сервісів?  
77.Для чого потрібні модулі? Скільки їх має бути в проєкті?  
78.Навіщо потрібні загальні модулі (shared)?  
79.Які переваги типізації в TypeScript?  
80.Які можливості TypeScript можна використовувати для типізації (тут мають на увазі інтерфейси, типи, enum тощо)?  
81.Яка різниця між інтерфейсом і класом?  
82.У чому різниця між інтерфейсом і абстрактним класом?  
83.Яка різниця між інтерфейсом і типом?  
84.Що таке RxJS? Як він використовується у фреймворку? Які компоненти фреймворку тісно пов’язані з ним?  
85.Чим відрізняються Observable і Promise?  
86.Для чого потрібні Subjects? Які типи Subjects існують?  
87.Як зробити кілька послідовних запитів до API за допомогою HTTP-сервісу і RxJS?  
88.Яка різниця між switchMap, concatMap, mergeMap?  
89.Як можна конфігурувати Angular-застосунок?  
90.Навіщо потрібні environment-файли? Коли їх краще не використовувати?  
91.У чому різниця між «розумним» (smart) і «дурним» (dumb) компонентами? У яких випадках застосовується кожен з них?  
92.У чому різниця між NgForm, FormGroup і FormControl і як їх застосовують для побудови форм?  
93.Навіщо потрібен і як працює async pipe?  
94.Як стежити за розвитком фреймворку? Яких відомих людей, пов’язаних з Angular, знаєте / читаєте?

**React**

95.Чи працювали з класовими компонентами? У чому їхня особливість?  
96.Які дані краще зберігати в стані компонента, а які передавати через пропси? Наведіть приклад.  
97.Чи ознайомлені з хуками? У чому їхні переваги? Чи доводилося робити свої і з якою метою?  
98.Чи ознайомлені з фрагментами та порталами? Навіщо вони потрібні?  
99.Коли й для чого використовують рефи?  
100.Які ви знаєте методи життєвого циклу компонента?  
101.В якому методі життєвого циклу компонента краще робити запити на сервер? Чому?  
102.В якому методі життєвого циклу компонента краще робити підписку і відписку від лістенера? Чому? Навіщо відписуватися?  
103.Чи був досвід роботи з контекстом? Коли його варто використовувати?  
104.У чому особливість PureComponent?  
105.Чи працював з мемоізованими селекторами (memoized selectors)? Для чого їх використовують і який принцип роботи?  
106.У чому бачите переваги бібліотеки React?  
107.Чому бібліотека React швидка? Що таке Virtual DOM і Shadow DOM?  
108.Навіщо в списках ключі? Чи можна робити ключами індекси елементів масиву? Коли це виправдано?  
109.В чому основна ідея Redux?  
110.Робота зі стилями в React.  
111.React — це бібліотека чи фреймворк? Яка різниця між цими двома поняттями.  
112.Чи можна використовувати jQuery разом з React? Чому так / ні?  
113.Що таке codemod?  
114.Чи доводилося налаштовувати проєкт React з нуля? За допомогою яких інструментів ви це робили?  
115.Перерахуйте всі бібліотеки, які використовували у зв’язці з React.  
116.Що найскладніше доводилося реалізовувати за допомогою React?

**Back-end**

117.Що таке REPL?  
118.Що таке streams в Node.js?  
119.Що таке middleware?  
120.Для чого використовують функцію setImmediate?  
121.Навіщо потрібен app.param() в express?  
122.Що таке token based authentication?

**Бази даних**

123.Напишіть простий запит для обчислення трьох авторів, у яких найбільше книг.  
124.Напишіть запит, який вибирає останні три коментарі для конкретного користувача для двох таблиць: коментарі та користувачі.  
125.Спроєктуйте просту схему бази даних для бібліотеки.  
126.Для чого використовують SQL-оператор HAVING?  
127.Навіщо використовують SQL-оператор LEFT JOIN?  
128.Чим відрізняється embed- від reference-зв’язку в MongoDB?  
129.В одному проєкті програмісти зберігають дані в MongoDB-колекції коментарів, використовуючи такі типи даних (дивіться нижче). Що поганого в цьому рішенні?

id: ObjectID

text: string

author\_id: string

created\_at: Date

130.У проєкті знадобилося внести зміни в структуру таблиць, додати кілька полів і індекси. Як програмісти будуть робити це на продакшені?

**Інструменти**

131.Щоразу, коли ви робите pull, чомусь трапляється конфлікт в останньому рядку в усіх файлах, які ви редагували. Що відбувається?  
132.Що робить команда git fetch?  
133.Які git hygiene підходи ви знаєте?  
134.Що таке CI/CD? Для чого це потрібно?

**Практичні завдання**

135.Розкажіть, які є способи копіювання простого об’єкта типу obj = {a: 1, b: 2, c: 3}  
136.Напишіть deep clone для об’єкта.  
137.Назвіть різні способи, як поміняти місцями значення двох змінних.  
138.Менеджер попросив у задачі поміняти статуси з «active, inactive» на «active, removed», але в коді фігурують тільки цифри й незрозуміло, який статус відповідає якій цифрі. Як допомогти майбутнім програмістам не лізти в документацію за кодом? Питання ставлять на конкретному прикладі з кодом.  
139.Необхідно зробити мініпроєкт — список користувачів з формою створення/редагування користувача:

Для зберігання користувачів використовуйте Firebase (це безкоштовно).

Для стилізації використовуйте Bootstrap.

Мінімальний набір полів користувача:

ім’я;

прізвище;

електронна пошта;

телефон (у форматі +380 (XX) XXX-XX-XX);

дата народження;

буде плюсом — додавання аватара та можливість crop-картинки.

Список користувачів повинен мати можливість фільтрації та пагінацію.

Проєкт має містити README-файл з кроками для запуску.