|  |  |  |
| --- | --- | --- |
| **姓名：李宜修**  ***Name:LEE I HSIU*** | **日期：2025/3/20**  ***Date:*** | **应征职位：Web Developer**  ***Position Applied*** |

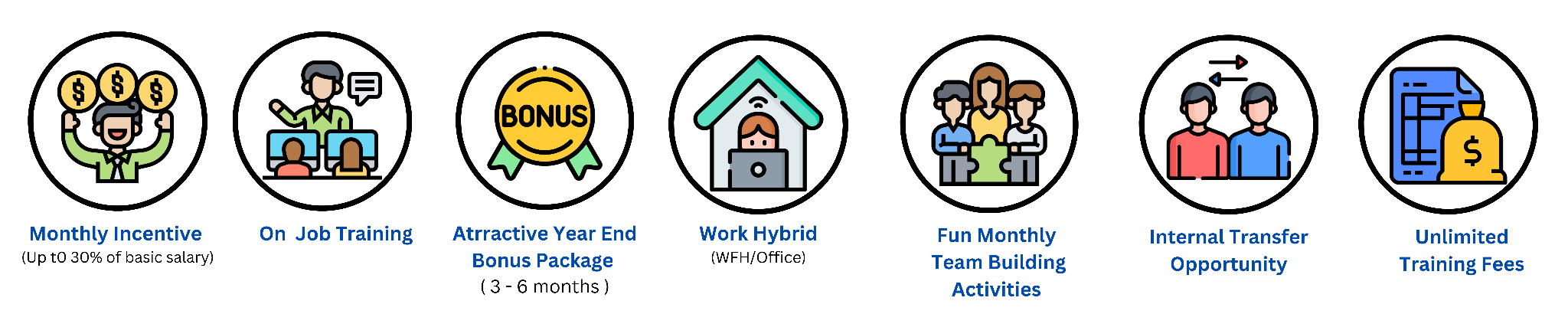
Dear Applicant,

Thank you for expressing your interest in our company. Please start with filling up this questionnaire for your job application and the interview process, job benefits have shown below for your reference too. Please do not hesitate to contact the HR team or recruiter if there’s any assistance needed. Again, appreciate your interest and we look forward to hearing from you!

|  |
| --- |
| **Interview Process :** |

****

|  |
| --- |
| **Job Benefits:** |



| **Personal Details** | | |
| --- | --- | --- |
| **学历** | ***2013/9-2017/7*** | ***东南大学电子科学与技术，本科毕业*** |
|  |  |
| **背景** | ***中国台湾*** | ***江苏省南京市雨花台区贾西中海城南公馆2期2栋4单元3007*** |
| ***生日*** | ***1995/4/18*** |
| **薪资** | **目前薪资** | ***22K \* 14~16 (RMB)*** |
| **期待薪资** | ***持平即可，20-30K*** |

|  |  |  |
| --- | --- | --- |
| **Work Experience ( Last 3 Jobs - Start from the latest job)** | | |
| ***2017/12～至今*** | ***华为*** | ***职位:软件开发工程师*** |
| 工作内容：  **2021/6-至今**：作为前端团队Leader带领团队（5人）交付以下内容： 1. “我的华为”微信、支付宝两个端侧的小程序，同时看护前端代码架构及安全送检事宜。使用技术栈包括vue, uniapp, taro等。 2. 负责跨端移动端H5页面-“通用报名”页面开发工作，交付页面应用于“我的华为APP”，“运动健康APP”等，消费者官网以及“我的华为”微信/支付宝小程序。主要使用第三方开源框架taro二次封装跨端引擎打造一套代码多端运行的页面，实现应用在以上多个端侧。 3. 负责终端IT内部PC平台，交付及开发工作，其平台面向华为内部运维、运营人员，可实时编辑我的华为app、小程序的首页装修、后台配置项等，动态地在平台上修改内容，即可实时运营2C端内容。 4. 使用React, Nextjs, 构建前端小程序工作台，node.js+mongodb构建express后端服务，miniprogram-ci工具，通过命令行执行打包、预览命令，一键上传至工作台，管理各环境各分支版本，用户可直接扫码体验，极大提升开发效率。  **2020-2021**: 负责IT内部低码平台：“构建器”的项目交付与开发，主要负责：完成低码平台的构建开发工作（Vue），开发构建器与渲染器功能，通过拖拉拽的简单操作，快速构建低保页面  **2019-2020**: 负责带领团队（4人）交付华为天猫二楼小程序，应用于天猫APP端侧，APP进入店铺页面，下拉选项后进入小程序，其内容包括商品选购，AR试戴等。使用技术栈包括threejs，vue，uniapp，阿里云开发等。  **2017-2019**: 主要使用Java语言开发内部IT平台的后端服务，使用的框架包括spring boot, tomcat, kafka等后端框架。主要对接运营商企业，交付项目主要为运营商机顶盒运营内容的编辑平台后端服务，运维可通过平台编辑上架影视内容来随时改变机顶盒呈现给消费者的运营内容。 |
| **离职原因：（目前仍在职）** | | |
| ***(YYYY/MM～YYYY/MM)*** | ***公司名字*** | ***职位:*** |
| ***工作内容：*** |

|  |
| --- |
| **For Web Developer** |
| **NodeJS**   * **请解释 NodeJS 的事件循环机制，以及为什么它适合处理 I/O 密集型任务**   **A: NodeJs的事件循环机制主要由两部分组成，事件队列和线程池。事件循环机制就是在这两者之间去判断他到底要执行任务还是等待延迟。比如当线程池内有一个任务完成了需要执行回调，这个回调会被调配到事件队列，当调用栈里没有其他任务时，就会将队列中的回调加入到调用栈中，得到执行。这样即使在单线程的JS环境中，也可以高效的处理并发任务。**  **Nodejs使用非阻塞I/O模型，所以当发起IO操作时，不会阻塞主线程，这个任务会经事件循环放到线程池中处理，同时还能处理其他请求，来达到高并发的效果。并且由于是单线程的，对比多线程的模型来说，不用考虑线程的切换或并发的复杂场景处理，资源的消耗也比较低。**   * **如何在 NodeJS 中设计和实现良好的错误处理和日志系统？**   **A: 首先对于错误的处理，我们可以分为几种**   1. **业务模块自身的错误处理，可以通过常用的try/catch，Promise.catch await.catch等来捕获错误，在回调中我们可以自定义错误的处理甚至上报到我们的日志系统。** 2. **全局的错误处理：即业务场景没处理到的一些异常，可以使用process.on来捕获全局的异常，比如uncaughtException 或 unhandledRejection都可以使用这个方法来捕获**   **那像现在很多开源的框架，自身都有统一的错误处理方法，比如Express就可以创建一个全局的错误处理handler，传到express app对象中来使用。**  **设计一个日志系统可以从以下几个点考虑：**   1. **日志的级别：error.warning,info,debug等** 2. **日志的格式定义和存储：直接使用json格式，定义报文的格式比如{id, dateTime, payload}。存储可以用fs来写文件。** 3. **日志系统区分环境：比如生产环境，不应有debug级别的日志产出。** 4. **避免敏感个人信息的日志上报。**   **可以考虑设计一个SDK，例如 class Logger(), 可以在constructor方法里面完成对日志区分环境、日志格式定义，甚至添加全局错误处理的方法（process.on）。然后定义每个等级的方法，比如logger.error(‘foobar’)**   * **你使用过哪些 NodeJS 开发框架，各自有什么优缺点？**  1. **Express:简单易用且轻量，上手快很容易就能构建Web API。而且express的生态系统也很丰富。** 2. **Mongoose: 当时是用来在接口中操作mongodb数据库，是一个ODM库。提供了schema定义数据库表格式的一种写法，也有很多中间件middleware的功能（比如用来实现一个文件存储系统）。自带的api简化了mongodb的操作** 3. **Nextjs：（不知道这算不算，但他也可以开发api）是在写React时用到的，一个全栈的框架，既可以做前端的React开发，也可以通过api router定义后端API生成接口，非常方便，又做到了前后端分离。而且支持SSR，typescript，生态社区也很丰富。** 4. **Prisma：与mongoose类似是一个ODM库，但项目里用到的PostgreSQL，因此用prisma来做一个对象关系映射的处理。他除了可以在代码层面定义schema，还可以使用脚手架工具来解析这个schema并在数据库里创建表、迁移等操作。深度集成了Typescript，也简化了PostgreSQL的数据库操作。**   **架构设计与性能**   * **如果让你设计整个 Admin 项目的架构，你会有什么样的考虑？**   **A: 我会从前后端分离的架构上去考虑，分别考虑前端和后端的设计，每个再去考虑他的技术选型和功能模块。**  **比如：**  **前端：**  **UI层，我们是使用什么框架实现（React,Vue,Svelte等），考虑到整个团队的技术栈。**  **UI组件库：开源的组件库较多，如果业务需求上定制的UI效果比较多，可以考虑例如daisyUI,shadcn等。或者用现成的elementui,bootstrapcss。**  **国际化：这个就看业务需求有没有国际化的要求了，框架的生态一般都用自己的i18n库**  **数据层：状态管理，比如React里的redux或Vue里的vuex等，可以用来管理全局数据层。**  **埋码：有没有前端埋码的需求，例如GA等**  **后端：**  **从技术选型开始，是使用后端语言（JAVA）还是前端（Expressjs还是KOA等），同样考虑到团队的技术栈。**  **数据库：数据库的选型，mongodb/postgreSQL/MySQL这些，包括要不要增加ORM工具（prisma/mongoose）**  **架构方面分为：网关层、数据层、登录鉴权层、日志监控层**  **网关层：可以考虑在后端系统的最开始添加一个gateway层，用来处理转发、限流等场景**  **数据层：就是使用ORM工具来实际对接数据库的API层，根据业务逻辑执行SQL的操作。**  **登录鉴权层：专门来处理鉴权场景，JWT、CSRF这些认证的能力。**  **日志：比如上面提到的日志工具，也需要为后台系统增加一个日志上报工具，专门处理后台的日志收集。包括当系统出现异常时的一个监控能力。**  **（可选）最后可以在添加一个swagger工具之类的，自动生成API文档，加快开发效率。一个简易的架构图如下：**     * **如果一个系统的页面加载时间过长，你会如何排查并优化性能？**   **首先一些常用的性能工具：**   1. **可以通过Chrome devTools的Performance/Lighthouse面板，开启录制加载页面，可以得到一个详细的报告，可以针对性的做各项指标的优化。或者使用比如Unlighthouse CLI工具来做一个google lighthouse的分析。** 2. **Web Vitals是一个小插件，可以监听页面上一些指标（FCP、LCP等），而且会直接打印出相关的元素。让你更快的找到有问题的元素。**   **这里的场景是页面加载时间过长，因此可以从页面包大小、网络请求、HTML渲染几个方面去考虑**   1. **看阻塞点是否在包的大小上，可以通过压缩js(webpack terser)，减少静态图片（部署到CDN），Tree shaking这几个方法来优化包的大小。如果可以作分包的话也可以考虑代码分割。** 2. **页面的网络请求过多时也会阻塞页面的加载，如果页面图片较多，可以考虑懒加载。或者把一些小的图片拼成雪碧图来代替，减少网络请求。并且可以做组件级的懒加载，不在可视区域内部的组件就不需要渲染。** 3. **阻碍HTML渲染的一般都是script标签的加载导致的，如果是一些非关键资源，可以用async,defer来异步加载。** 4. **分析是不是业务接口慢导致的页面加载慢，那这时候就轮到后台要优化接口性能了：比如优化SQL查询，或添加缓存策略来缓存重复查询的结果，或者可以使用graphql这样的框架，按需获取数据。** 5. **如果UI框架本身支持SSR，也可以考虑使用SSR来重构页面加载，肯定会比CSR要优。**  * **对 CSR、SSG、SSR 渲染模式的理解。**   **这三个分别对应三种前端渲染模式：**  **CSR：指的是客户端渲染，也是最常见的渲染模式。即浏览器下载并执行HTML和JS，在浏览器这一层动态的生成页面的内容。好处是他是前后端分离的，即端侧来执行渲染操作，并且对单页应用是很友好的。像使用Vue的应用大部分都是CSR，毕竟Vue他推崇的就是构建单页应用。**  **SSG：指的是静态站点生成的模式，像vitepress,vuepress这种，在构建时，他就直接生成静态的html文件。包的产物就都是html，那浏览器可以直接加载html，无需额外的渲染了。好处是他的加载速度肯定会比CSR更快，而且由于内容都在html上，浏览器可以抓取到你的内容，让他更容易被搜索到（SEO）。缺点就是整个是静态的，所以使用场景比较少，没有服务端的逻辑。我只在vitepress,vuepress这样的静态文档生成框架上使用过SSG。**  **SSR：SSR应该是现在最推崇的一种渲染方式，他把动态生成HTML的过程让服务器来执行，服务器直接返回HTML，浏览器可以直接渲染。就像SSG，但是他的内容由服务器来生成，所以是动态的。同时兼顾动态内容生成和速度快的优点，SEO友好。但是缺点在于对服务器的要求就更大了，而且开发的复杂度也会比以上两种要高。比如我在使用Nextjs时就经常会遇到组件是use client还是server side的考虑，但我若是写Vue完成用CSR就没这些考虑。**  **Other**   * **在前端领域，你最擅长的是什么？**   **个人认为我目前最擅长的应该是跨端的Web应用，多年的小程序经验积累下来，对小程序的开发也越来越熟悉。懂得如何处理跨端遇到的兼容性、性能问题等。并且因为我有后端的开发经验，设计整个架构方案涉及后端时更容易站在后端的角度来考虑，让他们更能接受我提出的方案。**  **从前端三大框架来考虑最擅长的就是Vue，后端的语言则是nodejs > java。**   * **对自己未来 1-3 年的规划**   **我认为未来3年可以参与一些更大的工程级别的复杂项目，不局限于前后端，因为近3年做的跨端项目量级其实不是很大，代码难度我个人认为算中等，更多的是在处理兼容性、性能问题等等。希望能跳脱舒适区，尝试一些有挑战性的项目。角色上还是以开发为主，如果需要带团队（有经验），我也不希望自己纯做项目管理，要保持自己的开发能力，至少让别人知道你是个懂技术的Leader。**  **开发能力上面继续朝前端能力精进多尝试一些别的框架Nextjs,svelte等，后端尽可能使用nodejs。并且不断学习如何让AI协助我开发，提升我的开发效率，自学python，AI基础等，看如何把AI实际应用到工作项目中。** |
| **🅰️:** |

**General questions:**

1. Would you prefer a position solely focused on management, or are you open to a role that primarily involves hands-on work?

🔺Your answer:

1. Do you need a work pass for work ?

🔺Your answer:

1. What is your notice period? Would you like to request any additional holidays while transitioning to a new job?

🔺Your answer:

1. On a scale of 1 to 10, please rate your proficiency in Chinese and English for (1) listening and speaking, and (2) reading and writing.

*This information is solely for informational purposes, and high scores are not necessarily required for most positions.*

🔺Your answer: Listening and Speaking :8 Reading and Writing: 9

1. What are the reasons motivating you to seek new job opportunities?

🔺Your answer:

1. Are you currently actively searching for new opportunities? Are you at the initial stages of your job search or are you in the final stages of evaluating offers?

🔺Your answer:

1. Compared to larger corporations, job roles at Pixl are often characterized by greater comprehensiveness and a wider scope. We encourage individuals to take on bigger responsibilities if they are willing to do so, as it allows for quick personal growth. Are you comfortable and open to working in such a challenging environment?

🔺Your answer:

1. How many years of experience do you have in the position that you’re applying for?

🔺Your answer

1. Have you collaborated or worked with individuals from overseas? If your answer is affirmative, please elaborate on your experience or share any relevant details. If not, would you be interested and willing to explore such opportunities?

🔺Your answer