## EMP CODE: 2726

**CANDIDATE NAME: Rashmi Parmar**

Designation: Sr Software Engineer

Company: Radix Web

**Summary**

A professional with 2 years of technical experience in Web Development using .NET Technologies with Angular. The Key skillset consists of technologies such as c#, .NET, jQuery, JavaScript, Angular, MS SQL Server, MVC, HTML, CSS 3, Tailwind and Bootstrap.

**Skills Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Beginner (1)** | **Competent (2)** | **Advanced (3)** | **Expert (4)** |

|  |  |
| --- | --- |
| **Technical / Functional Skills** | **Proficiency** |
| **Development & Deployment** | |
| * HTML5 | Advanced |
| * CSS3 | Advanced |
| * JavaScript | Competent |
| * TypeScript | Advanced |
| * MVC | Beginner |
| * Web API | Advanced |
| * Angular 13 | Advanced |
| * Entity Framework | Advanced |
| * Dotnet Core | Advanced |
| * SQL Server | Advanced |
| * AWS | Beginner |
| **General Areas** | |
| * Agile / Scrum Practice | Competent |
| * Experience with Atlassian Jira | Advanced |
| * GIT | Competent |
| **Soft Skills** | |
| * Communication | Advanced |
| * Client Handling including | Advanced |
| * Team / Individual Contributor | Advanced |

**Technical Highlights**

* Knowledge of OOPs Fundamentals
* Knowledge of UI Design, Development & Deployment
* Knowledge of Unit Testing.
* Experience working on various Microsoft .NET technologies.
* Basic knowledge of cloud fundamentals like Azure and AWS

**Project Experience**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** | **Payment Gateway** | **Duration** | **Months** |
| **Role** | Developer | **Team Size** | **6** |
| **Tools** | Angular 13, .Net Core 6.0, Entity Framework, Visual Studio Code 2019, AWS, SMTP, Twilio | **Database** | MS SQL Server |
| **Project Highlights** | This project is about Payment Gateway, I have created this project during the training period with a user-friendly interface accessible across desktop platforms, user can experience seamless financial interactions. And all transactions are managed by the database. | | |
| **Contribution** | Design, Develop | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.** | **Onboard Domain** | **Duration** | **5 Months** |
| **Role** | Developer | **Team Size** | **14** |
| **Tools** | Angular, .Net Core 6.0, Entity Framework, Visual Studio Code 2019, AWS | **Database** | MS SQL Server |
| **Project Highlights** | This project is about an onboarding service. This platform prioritizes a seamless onboarding experience, minimizing friction and maximizing user understanding. Through interactive elements and progressive disclosure, it’s ensured a positive first impression, fostering increased user retention. | | |
| **Contribution** | Design, Development of enhancement, Bug solving, Unit testing, Communication | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.** | **Recruitment Software** | **Duration** | **1 months** |
| **Role** | Developer | **Team Size** | **5** |
| **Tools** | MVC (4.7), Azure | **Database** | MS SQL Server |
| **Project Highlights** | This project is about Recruitment process, The agency has multiple clients and candidates. The candidate may have different skills and availability. The agency generates invoices from clients as per the number of candidates and several timesheets. The candidate is hired as per booking. The timesheet is filled for a particular period mentioned in the booking. The timesheets are approved by the user for each candidate. There may have been a change in hour rate and hours. The final payroll process is being done to generate pay for each candidate. Each candidate is provided a payslip and a notification through email. The tax is calculated as per the Tax code for each candidate. | | |
| **Contribution** | - Support and maintenance | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **4.** | **Real Estate Site** | **Duration** | **1 year** |
| **Role** | Developer | **Team Size** | **9** |
| **Tools** | Angular 13, .Net Core 6.0, Entity Framework, Visual Studio Code 2019 | **Database** | MS SQL Server |
| **Project Highlights** | This project is about Real Estate Website, it typically incorporates a user-friendly interface for property seekers to browse listings, employing advanced search functionalities for specific criteria. Features like virtual tours, high quality images, and detailed property descriptions enhance user experience. The incorporation of data analytics and machine learning algorithms can enhance user engagement and provide personalized recommendations. | | |
| **Contribution** | Played a key role in fine-tuning SQL Server queries, supporting role for Dot Net | | |

**Academic Qualifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course** | **Board / University** | **Year Of Passing** | **Grade** |
| **BE(IT)** | LDRP – ITR Gandhinagar | 2022 | 6.80 CGPA (out of 10) |
| **HSC** | State Board, Gujarat | 2018 | 60% |
| **SSC** | State Board, Gujarat | 2016 | 82.92% |

Interview Question & Answers

* Project Payment Gateway:

1. Can you describe your role and contribution in the Payment Gateway project?

* During the Payment Gateway project, my role primarily involved designing and developing the system. I contributed to creating a user-friendly interface accessible across desktop platforms and ensuring seamless financial interactions.

1. What technologies and tools did you use to develop the Payment Gateway?

* We utilized Angular 13 for front-end development, .Net Core 6.0 for backend functionality, Entity Framework for database interactions, Visual Studio Code 2019 for coding, and AWS for hosting. Additionally, we integrated SMTP for email notifications and Twilio for SMS notifications.

1. How did you ensure user-friendliness in the interface of the Payment Gateway?

* User-friendliness was a key focus in our project. We incorporated intuitive design principles and conducted user testing to refine the interface. Features such as clear navigation, concise messaging, and responsive design were implemented to enhance user experience.

1. Could you elaborate on how transactions are managed by the database in the Payment Gateway?

* In our Payment Gateway, transactions are managed by the MS SQL Server database. We designed robust database schemas to store transaction details securely. SQL queries and stored procedures were optimized for efficient retrieval and manipulation of transaction data.

1. What challenges did you encounter during the development of the Payment Gateway, and how did you overcome them?

* One challenge we faced was ensuring secure communication between different components of the system. We addressed this by implementing HTTPS protocols and encryption techniques. Additionally, we conducted thorough testing and code reviews to identify and rectify any vulnerabilities.

1. Can you explain the integration process of AWS, SMTP, and Twilio in the Payment Gateway project?

* AWS was used for hosting the application, SMTP for sending email notifications related to transactions, and Twilio for sending SMS notifications. We integrated these services seamlessly into our application using their respective APIs and SDKs.

1. What was the rationale behind selecting Angular 13 and .Net Core 6.0 for this project?

* Angular 13 and .Net Core 6.0 were chosen for their robustness, scalability, and community support. Angular provided a comprehensive framework for building dynamic single-page applications, while .Net Core offered high performance and cross-platform compatibility for backend services.

1. How did you handle security considerations, such as data encryption and user authentication, in the Payment Gateway?

* Security was a top priority in our Payment Gateway project. We implemented encryption algorithms to secure sensitive data such as payment details and user credentials. Additionally, we enforced strict authentication measures, including multi-factor authentication, to prevent unauthorized access.

1. Can you discuss any optimizations or performance enhancements you implemented in the Payment Gateway?

* To optimize performance, we employed caching mechanisms to reduce database queries and implemented asynchronous processing for time-consuming tasks. We also conducted performance testing and fine-tuned system configurations to achieve optimal responsiveness.

1. How did you manage teamwork and collaboration within your team of six developers during the project?

* Teamwork and collaboration were essential for the success of our project. We utilized agile methodologies such as Scrum, daily stand-up meetings, and task tracking tools to coordinate efforts and ensure timely delivery. Regular communication and knowledge sharing among team members facilitated smooth progress throughout the project lifecycle.
* Onboarded Domain:

1. Can you elaborate on your role and contributions in the Onboard Domain project?

* In the Onboard Domain project, my role encompassed various responsibilities including design, development of enhancements, bug solving, unit testing, and communication. I contributed to creating a seamless onboarding experience for users by designing intuitive interfaces and implementing functionalities to enhance user understanding.

1. Which tools and technologies did you utilize for the development of the onboarding service?

* For the development of the Onboard Domain project, we utilized Angular for front-end development, .Net Core 6.0 for backend services, Entity Framework for database interactions, Visual Studio Code 2019 for coding, and AWS for hosting the application. The database used was MS SQL Server.

1. How did you ensure a seamless onboarding experience for users in the Onboard Domain project?

* A seamless onboarding experience was achieved through careful design considerations and user-centric development practices. We prioritized intuitive navigation, clear instructions, and progressive disclosure to guide users through the onboarding process smoothly.

1. Could you explain the importance of interactive elements and progressive disclosure in fostering a positive first impression for users?

* Interactive elements and progressive disclosure play a crucial role in making a positive first impression on users. Interactive elements engage users and provide them with a sense of control, while progressive disclosure gradually reveals information, reducing cognitive overload and enhancing comprehension.

1. What challenges did you face during the development of the Onboard Domain project, and how did you address them?

* Challenges encountered during the Onboard Domain project included ensuring compatibility across various devices and browsers, optimizing performance for scalability, and addressing security concerns. We addressed these challenges through thorough testing, optimization techniques, and adherence to security best practices.

1. How did you utilize AWS in the Onboard Domain project, and what specific services did you rely on?

* AWS was utilized for hosting the Onboard Domain application. We relied on services such as EC2 for virtual server hosting, S3 for storing static assets, and RDS for managing the MS SQL Server database. Additionally, AWS Lambda was used for serverless computing tasks.

1. What were the reasons behind selecting Angular and .Net Core 6.0 for this project?

* Angular and .Net Core 6.0 were chosen for their robustness, scalability, and extensive community support. Angular provided a comprehensive framework for building dynamic single-page applications, while .Net Core offered high performance and cross-platform compatibility for backend services.

1. How did you ensure data integrity and security in the MS SQL Server database used for the Onboard Domain project?

* Data integrity and security were paramount in the Onboard Domain project. We implemented strict access controls, encrypted sensitive data, and regularly updated security measures to protect against potential threats. Additionally, we conducted regular database backups to ensure data resilience.

1. Can you discuss any performance optimizations or scalability enhancements implemented in the Onboard Domain project?

* To optimize performance and ensure scalability, we employed caching mechanisms to reduce database queries and implemented asynchronous processing for time-consuming tasks. Additionally, we monitored system performance metrics and fine-tuned configurations as needed to accommodate increasing user loads.

1. How did you manage communication and collaboration within your team of 14 developers during the project?

* Communication and collaboration within our team of 14 developers were facilitated through regular meetings, clear task assignments, and collaborative tools such as Slack and Jira. We fostered an environment of open communication and knowledge sharing to ensure alignment and progress towards project goals.
* **Recruitment Software**:

1. Can you describe your role in the Recruitment Software project and the specific tasks you were responsible for during the development phase?

* In the Recruitment Software project, my role primarily revolved around support and maintenance. I was responsible for addressing any issues that arose post-development, ensuring the smooth functioning of the software, and implementing updates or enhancements as needed based on user feedback.

1. What tools and technologies did you utilize to develop the Recruitment Software?

* For the Recruitment Software project, we utilized MVC 4.7 for the development of the application and Azure for hosting. The database used was MS SQL Server, providing a robust and scalable platform for storing and managing recruitment-related data.

1. Could you walk us through the key functionalities and processes of the Recruitment Software, such as candidate management, invoicing, timesheet management, and payroll processing?

* The Recruitment Software facilitates various aspects of the recruitment process, including candidate management, invoicing, timesheet management, and payroll processing. Clients submit requirements for candidates, and the agency matches suitable candidates based on skills and availability. Invoices are generated based on the number of candidates and timesheets submitted. Timesheets are filled for specific periods mentioned in the booking and approved by users. The final payroll process calculates pay for each candidate, considering changes in hour rates and hours worked. Each candidate receives a payslip and email notification, with tax calculated as per their tax code.

1. How did you handle changes in hour rates and hours for candidates in the Recruitment Software?

* Changes in hour rates and hours for candidates are managed through the timesheet management functionality. Users can update the hour rates and hours worked for candidates within the system, and these changes are reflected in the payroll processing and invoice generation processes.

1. Can you explain the process of generating invoices for clients based on the number of candidates and timesheets in the Recruitment Software?

* Invoices for clients are generated based on the number of candidates and timesheets submitted. The system calculates the total amount owed by each client based on the agreed-upon rates and hours worked by the candidates. Invoices are then generated and sent to clients for payment.

1. What measures did you take to ensure accurate tax calculation for each candidate based on their tax code?

* Tax calculation for each candidate is based on their tax code, which is stored in the system. The software automatically calculates tax deductions based on the candidate's tax code and generates accurate payslips reflecting the net pay after tax deductions.

1. How were candidate payslips generated and distributed in the Recruitment Software, and what role did email notifications play in this process?

* Candidate payslips are generated within the software and distributed to candidates electronically. Email notifications are sent to candidates along with their payslips, providing them with a summary of their earnings and deductions for the specified period.

1. What challenges did you encounter during the development of the Recruitment Software, particularly in terms of handling multiple clients, candidates, and timesheets?

* Challenges during the development of the Recruitment Software included handling multiple clients, candidates, and timesheets efficiently, ensuring accurate calculation of invoices and payroll, and implementing robust security measures to protect sensitive candidate and client data.

1. How did you ensure data integrity and security in the MS SQL Server database used for the Recruitment Software?

* Data integrity and security were ensured in the MS SQL Server database by implementing strict access controls, encryption techniques for sensitive data, and regular backups to prevent data loss in case of system failures or breaches.

1. As you were primarily involved in support and maintenance, can you discuss any specific issues you addressed and how you resolved them during the project lifecycle?

* In my role focused on support and maintenance, I addressed various issues such as system errors, performance optimizations, and user interface enhancements. These issues were resolved through thorough troubleshooting, collaboration with the development team, and timely implementation of solutions to minimize downtime and disruption to users.
* **Real Estate Site**:

1. Can you describe your role in the Real Estate Site project and the specific tasks you were responsible for during development?

* In the Real Estate Site project, I served as a developer and played a key role in fine-tuning SQL Server queries to optimize database performance. Additionally, I provided support for Dot Net aspects of the project, collaborating with the development team to ensure the successful implementation of features and functionalities.

1. What tools and technologies did you utilize to develop the Real Estate Site?

* For the Real Estate Site project, we utilized Angular 13 for front-end development, .Net Core 6.0 for backend services, Entity Framework for database interactions, and Visual Studio Code 2019 for coding. These tools and technologies provided a robust foundation for building a modern and responsive real estate website.

1. What were the key features and functionalities of the Real Estate Site, particularly in terms of user interface, property listings, and search capabilities?

* The Real Estate Site offered a user-friendly interface for property seekers to browse listings, incorporating advanced search functionalities for specific criteria such as location, price range, and property type. Features like virtual tours, high-quality images, and detailed property descriptions were integrated to enhance the user experience and provide comprehensive information about listed properties.

1. How did you incorporate advanced search functionalities to allow users to search for properties based on specific criteria?

* Advanced search functionalities in the Real Estate Site allowed users to filter property listings based on various criteria, enabling them to find properties that match their specific preferences and requirements. Users could refine their search results by specifying parameters such as property features, amenities, and proximity to amenities.

1. Could you explain how virtual tours, high-quality images, and detailed property descriptions were integrated into the Real Estate Site to enhance user experience?

* Virtual tours, high-quality images, and detailed property descriptions were integrated into the Real Estate Site to provide users with a comprehensive view of listed properties. Virtual tours allowed users to explore properties remotely, while high-quality images and detailed descriptions provided additional context and information to help users make informed decisions.

1. How did you contribute to fine-tuning SQL Server queries in the Real Estate Site, and what strategies did you employ to optimize database performance?

* I contributed to fine-tuning SQL Server queries in the Real Estate Site to optimize database performance. This involved analyzing query execution plans, identifying bottlenecks, and implementing optimizations such as index tuning and query restructuring to improve overall database efficiency and responsiveness.

1. Can you discuss any role you played in supporting Dot Net aspects of the Real Estate Site development?

* In addition to fine-tuning SQL Server queries, I provided support for Dot Net aspects of the Real Estate Site development. This included collaborating with the development team to troubleshoot issues, implement new features, and ensure the smooth integration of front-end and back-end components.

1. Did you explore the incorporation of data analytics and machine learning algorithms in the Real Estate Site, and if so, how were they utilized to enhance user engagement and provide personalized recommendations?

* While data analytics and machine learning algorithms were not explicitly incorporated into the Real Estate Site during my involvement, I recognize their potential to enhance user engagement and provide personalized recommendations. Integrating these technologies could involve analyzing user behavior and preferences to deliver tailored property recommendations and improve the overall user experience.

1. What challenges did you encounter during the development of the Real Estate Site, and how did you overcome them?

* Challenges encountered during the development of the Real Estate Site included optimizing performance for large datasets, ensuring compatibility across different devices and browsers, and implementing robust security measures to protect user data. These challenges were addressed through thorough testing, performance optimization techniques, and adherence to best practices in web development.

1. How did you ensure the scalability and performance of the Real Estate Site, particularly considering potential increases in user traffic and property listings over time?

* To ensure the scalability and performance of the Real Estate Site, we implemented strategies such as caching frequently accessed data, optimizing database queries, and leveraging cloud-based infrastructure for scalability. Additionally, we regularly monitored system performance metrics and conducted load testing to identify and address any potential bottlenecks or scalability issues proactively.