**GBTC - Giving Back to Community Platform**

**Requirements Document**

**1. Introduction**

GBTC (Giving Back to Community) is a web-based platform aimed at facilitating knowledge sharing, skill development, and networking within the software industry community. The platform caters to both students aspiring to enter the software industry and professionals already employed in it. By providing a range of features such as blogs, live streams, project showcases, interview experiences, and referrals, GBTC aims to create a vibrant community where members can learn, teach, and support each other.

**2. User Roles**

GBTC supports two types of users:

* **Students:** Individuals currently pursuing education or seeking entry into the software industry.
* **Employed Professionals:** Individuals already employed in the software industry who wish to give back to the community and support aspiring professionals.

**3. Functional Requirements**

**3.1 Registration and Authentication:**

* Users can register with GBTC by providing basic information such as name, email, and password.
* Upon registration, users can authenticate themselves using their credentials to access the platform.

**3.2 User Profile:**

* Each user has a profile page where they can view and update their personal information.
* Users can choose their role (student or employed professional) during registration and update it later if necessary.

**3.3 Student Features:**

* Read Blogs: Students can access and read blogs covering various topics such as career roadmap, technical skills, and industry trends.
* Attend Live Streams: GBTC hosts live streams where industry experts share insights, conduct workshops, and answer questions.
* Publish Blogs: Students can contribute to the platform by publishing their own blogs on topics of interest.
* Showcase Projects: Students can showcase their personal projects to receive feedback and recognition.
* Coding Challenges: GBTC hosts weekly coding challenges where students can participate, solve problems, and compare their solutions with others.
* Project Collaboration: GBTC facilitates project collaboration among students by matching individuals with complementary skills and interests.
* Industry Insights Blog Series: GBTC publishes a series of industry insights blogs written by experienced professionals.

**3.4 Employed Professional Features:**

* All Student Features: Employed professionals have access to all features available to students.
* Host Live Streams: Professionals can host live streams on topics they are knowledgeable about, providing valuable insights and advice to the community.
* Give Referrals: Employed professionals can refer talented students for job opportunities within their organizations.
* Expert Panel Q&A Sessions: GBTC hosts monthly expert panel Q&A sessions where employed professionals from different domains answer questions related to their expertise.
* Referral Reward Program: Employed professionals can participate in GBTC's referral reward program.
* Leadership Workshops: GBTC offers leadership workshops designed specifically for employed professionals seeking to advance their careers.
* Remote Work Resources: GBTC provides resources and tips for professionals adapting to remote work environments.

**3.5 Interview Experiences:**

* Users can share their interview experiences, including technical questions, behavioral interviews, and company culture.
* Interviews can be posted anonymously or non-anonymously based on user preference.

**4. Non-Functional Requirements**

**4.1 Performance:**

* The platform should be responsive and scalable to handle concurrent user interactions.
* Page load times should be optimized to provide a seamless user experience.

**4.2 Security:**

* User data should be encrypted during transmission and storage to ensure privacy and security.
* Authentication mechanisms should be robust to prevent unauthorized access to user accounts.

**4.3 Usability:**

* The user interface should be intuitive and user-friendly, with clear navigation and informative feedback messages.
* Accessibility standards should be followed to ensure that the platform is usable by individuals with disabilities.

**4.4 Compatibility:**

* The platform should be compatible with a wide range of devices and web browsers to accommodate diverse user preferences.

**Design Document**

**1. Architecture Overview**

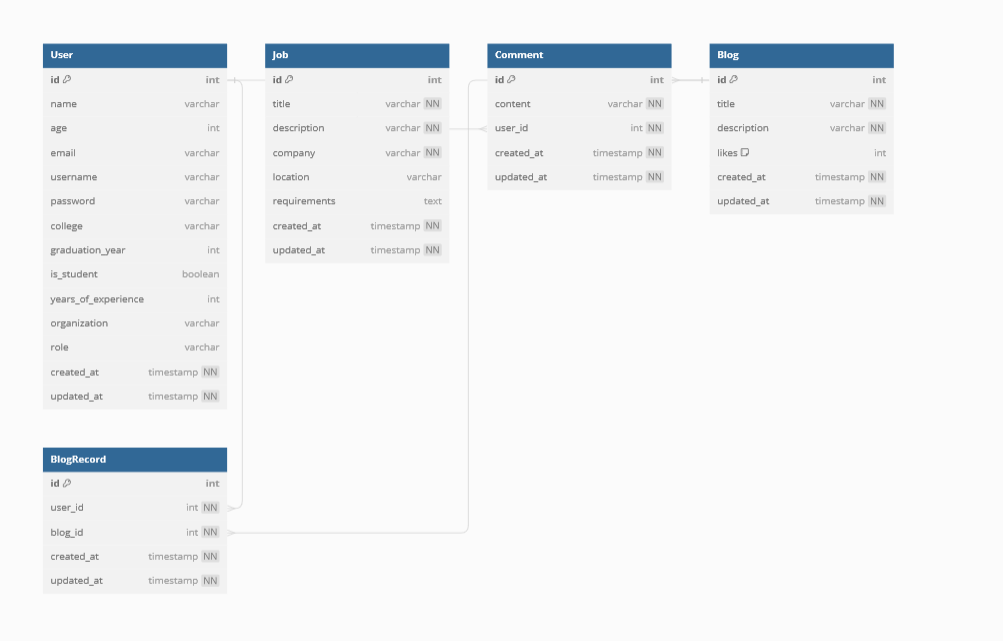
GBTC follows a modular architecture based on the MERN (MongoDB, Express.js, React.js, Node.js) stack:

* **Frontend:** Developed using React.js, the frontend interacts with the backend APIs to fetch data and render the user interface.
* **Backend:** Built using Express.js and Node.js, the backend provides RESTful APIs for user authentication, blog management, project showcases, live streams, referrals, and interview experiences.
* **Database:** MongoDB is used to store user data, blog content, project details, interview experiences, and other relevant information.

**2. Database Design**

The MongoDB database consists of the following collections:

* **Users:** Stores user information including name, email, password hash, role, and profile details.
* **Blogs:** Contains blog posts contributed by users, including title, content, author, timestamps, and tags.
* **Projects:** Stores project details such as title, description, technologies used, author, and timestamps.
* **Interview Experiences:** Stores interview experiences shared by users, including company, interview questions, difficulty level, author, timestamps, and anonymity flag.
* **Referrals:** Stores referral information including the referring user, referred user, organization, job position, and status.



**3. API Design**

GBTC backend provides the following RESTful APIs:

* **Authentication APIs:**
  + /api/auth/signup: Register a new user.
  + /api/auth/login: Authenticate user credentials and generate a JWT token.
  + /api/auth/user: Get current user details.
* **Blog APIs:**
  + /api/blogs: Get all blogs or create a new blog post.
  + /api/blogs/:id: Get, update, or delete a specific blog post by ID.
  + /api/blogs/tags/:tag: Get blogs filtered by tag.
* **Project APIs:**
  + /api/projects: Get all projects or create a new project.
  + /api/projects/:id: Get, update, or delete a specific project by ID.
* **Interview Experience APIs:**
  + /api/interviews: Get all interview experiences or create a new interview experience.
  + /api/interviews/:id: Get, update, or delete a specific interview experience by ID.
  + /api/interviews/anonymity: Set anonymity preference for interview experiences.
* **Referral APIs:**
  + /api/referrals: Get all referrals or create a new referral.
  + /api/referrals/:id: Get, update, or delete a specific referral by ID.
* **Live Stream APIs:**
  + /api/live-streams: Get all live streams or create a new live stream.
  + /api/live-streams/:id: Get, update, or delete a specific live stream by ID.

**4. Frontend Design**

The frontend is designed using React.js and styled with CSS or CSS frameworks such as Bootstrap or Material-UI. It consists of the following main components:

* **Navigation Bar:** Provides navigation links to different sections of the platform including blogs, projects, live streams, interview experiences, and referrals.
* **User Profile:** Displays user information and allows users to update their profile details.
* **Blogs Section:** Displays a list of blog posts with pagination and filtering options by tags.
* **Projects Section:** Displays a list of projects with filtering options by technologies used.
* **Interview Experiences Section:** Allows users to share their interview experiences and view others' experiences.
* **Referral Section:** Allows employed professionals to give referrals and students to view available job opportunities.
* **Live Streams Section:** Displays upcoming and past live streams with details and registration options.

**5. Deployment**

GBTC can be deployed on a cloud platform such as AWS, Google Cloud Platform, or Microsoft Azure. The frontend can be hosted using services like Amazon S3 or Netlify, while the backend can be deployed on platforms supporting Node.js applications like AWS Elastic Beanstalk or Heroku. MongoDB Atlas can be used for database hosting.

**6. Testing**

Unit tests can be written using testing frameworks such as Jest for frontend components and Mocha/Chai for backend APIs. Integration tests can be performed using tools like Cypress for end-to-end testing. Continuous integration and deployment (CI/CD) pipelines can be set up using platforms like GitHub Actions or Jenkins.

**7. Scalability and Performance**

GBTC should be designed for scalability to handle increasing user loads. This can be achieved by implementing caching mechanisms, load balancing, and horizontal scaling strategies. Performance optimizations should be applied at both frontend and backend levels to ensure fast response times and efficient resource utilization.

**8. Security**

GBTC should implement security best practices including encryption of sensitive data, input validation, protection against common web vulnerabilities such as cross-site scripting (XSS) and SQL injection, and secure authentication mechanisms using JWT tokens.

**9. Accessibility**

GBTC should adhere to web accessibility standards such as WCAG (Web Content Accessibility Guidelines) to ensure that the platform is usable by individuals with disabilities. This includes providing alternative text for images, keyboard navigation support, and semantic HTML markup.

**10. Maintenance and Support**

Regular maintenance and updates should be performed to keep the platform up-to-date with the latest technologies and security patches. A support system should be in place to address user inquiries, bug reports, and feature requests in a timely manner.

**11. Conclusion**

GBTC is designed to provide a comprehensive platform for knowledge sharing, skill development, and networking within the software industry community. By following the outlined design principles and implementing the suggested features, GBTC aims to create a vibrant and inclusive community-driven platform for users to learn, collaborate, and grow together.