



**Where Service Meets
You, Anywhere**

By: Anas, Amjad, Harshil, Aneeq, Majed

What Problems does BizReach Solve?

BizReach is a platform for individuals above the age of 16.

Provides clientele for Business owners and services for customers.

Main Solutions



Trust and Credibility



Convenience



Local Business Promotion

Process

- All Communication: Discord
- Initial Meetings at start of sprint
- GitHub Workflow
- Good decisions: Detailing workflow, Cleaning up code base
- Bad decisions: Neglecting UI



Technical Discussion - FERN Stack

Technical Items:



Chakra UI

Significant difference in appealing UI



Socket.io

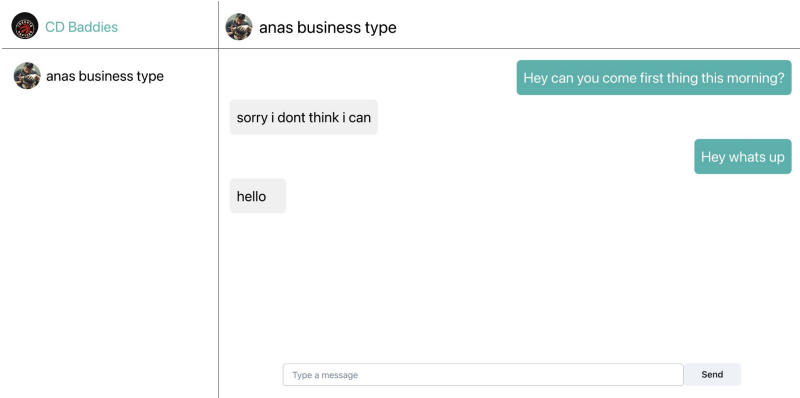
Enabling bi-directional communication between clients and businesses



Firebase

Real-time data synchronization for chats, posts & accounts

Technical Discussions



```
useEffect(() => {
  const socket = io('http://localhost:3000');
  socket.on('newMessage', (data) => {
    setChatMessagesData((prevChatMessages) => ({
      ...prevChatMessages,
      [data.senderId]: [
        ...(prevChatMessages[data.senderId] || []),
        { text: data.text, sender: { _path: { segments: ['', data.senderId] } } },
      ],
    }));
  });
  return () => {
    socket.disconnect();
  };
}, []);
```

A screenshot of a business profile form. On the left is a sidebar with navigation links: 'Personal Info', 'Business Info' (highlighted), 'Support', 'Privacy Policy', and 'Logout'. The main form area has the following sections:

- Business Name:** A text input field containing 'Cat Groomer'.
- Business Description:** A text input field containing 'I groom cats'.
- Rating:** A section showing a rating of 5, with a list of days from Monday to Sunday, each with a checkmark.
- Business Hours:** A section showing a list of days from Monday to Sunday, each with a time range (e.g., 'Monday 14:00 - 19:00').

At the bottom right of the form is a 'Save' button.

Software Architecture

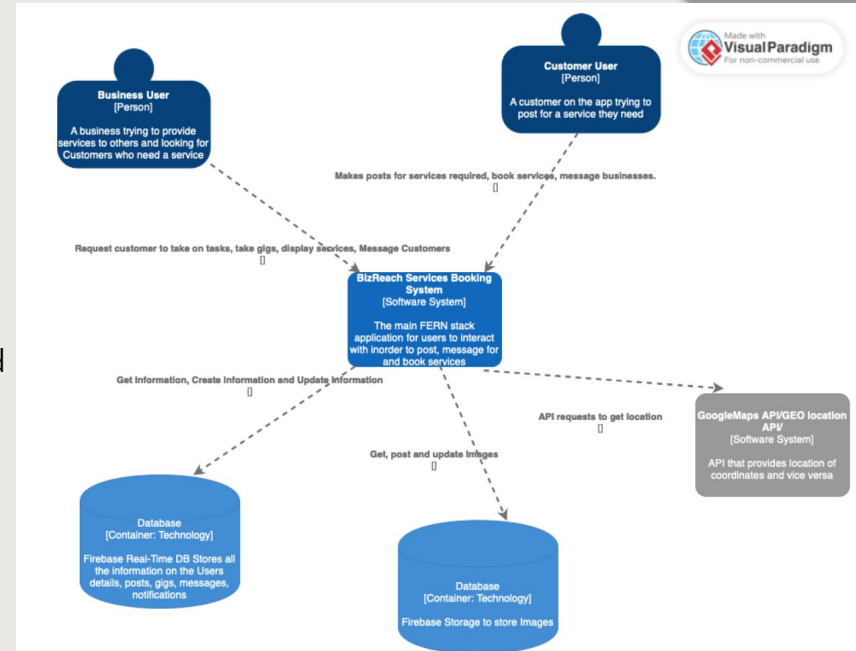
- Main components
- Responsibilities
 - Interaction
- Technology and tools used
- Challenges faced
- Software techniques used



Software Architecture

Main components/Responsibilities and connection

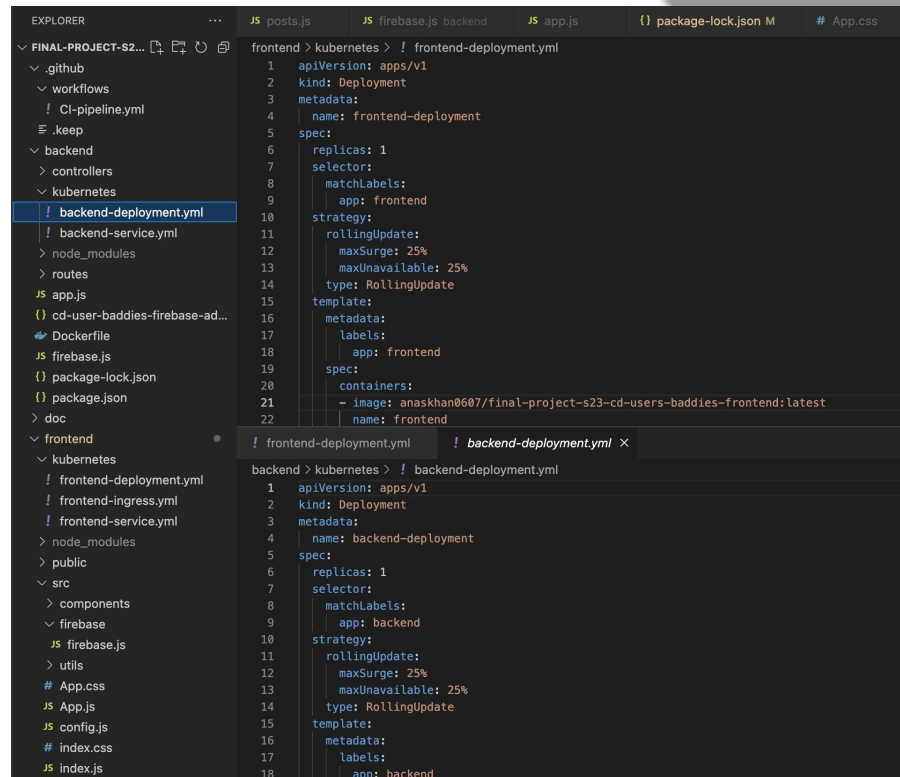
- React Frontend:
 - Intuitive user interface
 - Dynamic user experience
- Express.js Backend:
 - Intermediary between frontend and Firebase
 - Handles business logic and functionalities
- Node.js:
 - Powers the server environment
 - Efficient communication between frontend and backend
- Firebase:
 - Real-time database
 - Authentication and user management
 - Image storage
- Google Maps API:
 - Translates user-provided locations into actionable data



Software Architecture

Deployment

- Docker:
 - Created Docker files for backend and frontend
- Kubernetes:
 - Set up deployments and services
- Ingress:
 - Configured for frontend
- EasyDNS:
 - Host obtained for frontend
- Google Kubernetes Engine (GKE):
 - Hosting platform for scalability and availability



The screenshot shows a code editor with two tabs open: 'frontend-deployment.yml' and 'backend-deployment.yml'. The 'frontend-deployment.yml' tab is active, showing a Kubernetes Deployment manifest for the 'frontend' application. The manifest includes fields for 'apiVersion', 'kind', 'metadata', 'spec', and 'template'. The 'spec' section defines 'replicas: 1', 'selector', 'strategy' (RollingUpdate), and 'template' (metadata, labels, spec, containers). The 'containers' section defines the 'image' as 'anaskhan0607/final-project-s23-cd-users-baddies-frontend:latest' and the 'name' as 'frontend'. The 'backend-deployment.yml' tab is also visible, showing a similar manifest for the 'backend' application.

```
EXPLORER
... JS posts.js JS firebase.js backend JS app.js {} package-lock.json M # App.css
v FINAL-PROJECT-S2...
  v .github
  v workflows
  v CI-pipeline.yml
  v .keep
  v backend
    > controllers
    v kubernetes
      ! backend-deployment.yml
      ! backend-service.yml
      > node_modules
      > routes
      JS app.js
      {} cd-user-baddies-firebase-ad...
      Dockerfile
      JS firebase.js
      {} package-lock.json
      {} package.json
      > doc
  v frontend
    v kubernetes
      ! frontend-deployment.yml
      ! frontend-ingress.yml
      ! frontend-service.yml
    > node_modules
    > public
    v src
      > components
      v firebase
        JS firebase.js
      > utils
      # App.css
      JS App.js
      JS config.js
      # index.css
      JS index.js

frontend > kubernetes > ! frontend-deployment.yml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: frontend-deployment
5 spec:
6   replicas: 1
7   selector:
8     matchLabels:
9       app: frontend
10  strategy:
11    rollingUpdate:
12      maxSurge: 25%
13      maxUnavailable: 25%
14    type: RollingUpdate
15  template:
16    metadata:
17      labels:
18        app: frontend
19    spec:
20      containers:
21        - image: anaskhan0607/final-project-s23-cd-users-baddies-frontend:latest
22          name: frontend

! frontend-deployment.yml ! backend-deployment.yml x
backend > kubernetes > ! backend-deployment.yml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: backend-deployment
5 spec:
6   replicas: 1
7   selector:
8     matchLabels:
9       app: backend
10  strategy:
11    rollingUpdate:
12      maxSurge: 25%
13      maxUnavailable: 25%
14    type: RollingUpdate
15  template:
16    metadata:
17      labels:
18        app: backend
```


Software Architecture

Technology and Tools Used

- Firebase:
 - Real-time database and authentication
- Express.js and Node.js:
 - Backend communication
- React:
 - User interface development
- Docker:
 - Streamlined deployment
- Kubernetes:
 - Container orchestration
- GKE:
 - Hosting and scalability
- Socket.io:
 - Real-time communication implementation

Software Architecture

Challenges Faced

- Challenge: Real-time updates for posts and messages
- Solution: Implemented Socket.io for persistent connections

- Challenge: Restructuring codebase for better separation
- Solution: Enhanced separation of frontend and backend for maintainability

Software Architecture

Conclusion

- FERN stack choice aligned with project needs
- Components worked together seamlessly
- Deployment process ensured scalability and reliability
- Overcame challenges using innovative techniques
- Excited to demonstrate our application's architecture in action

Each Members' Contribution

- Anas: posts gigs flow, structuring code
- Amjad: Sign up, registration flow, profile page
- Harshil: Messaging feature, uploading pictures feature
- Aneeq: Reviews, Posts and Business features
- Majad: Discover Businesses, my gigs



Q & A