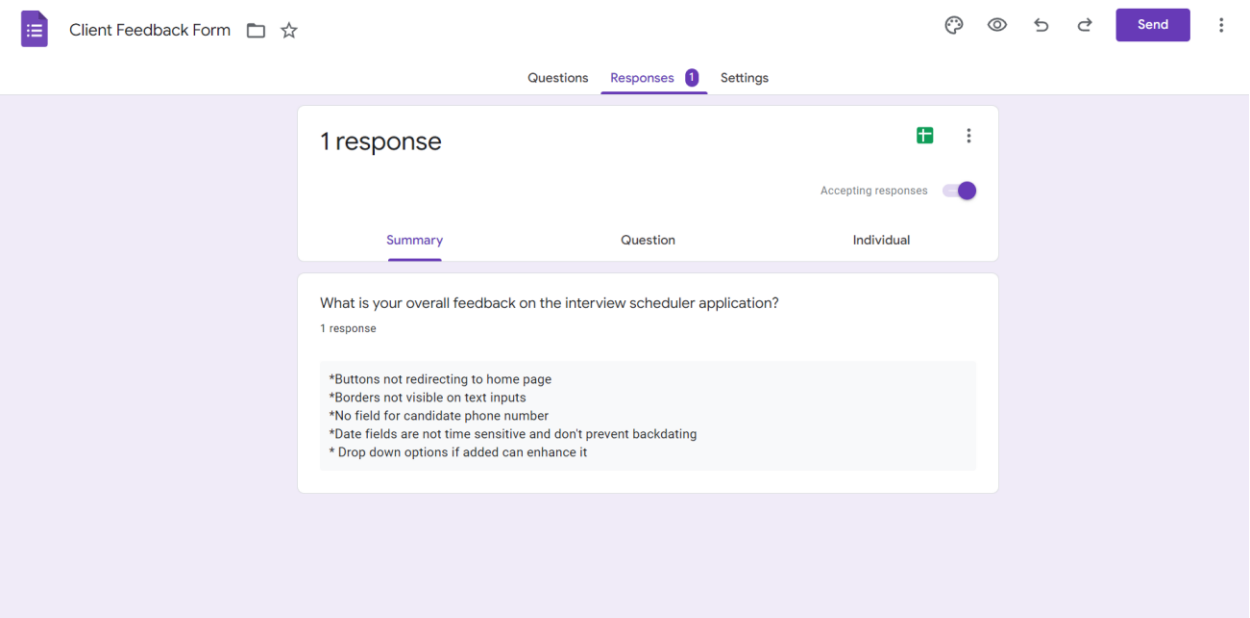
Appendix



Client feedback form response

Success criteria	**Testing plan**	**Results**
Ability to create users	Creating two separate users using firebase	Criteria Met
Ability to authenticate users to separate interview data	Logging in from two different accounts to ensure that each has different data	Criteria Met
Ability to remain logged into the application	Reloading page, closing window and closing browser to check if user is still logged in	Criteria Met
Ability to create positions that are stored in online database	Creating positions and checking for them in home page and on the server	Only partially met because of UI error
Ability to add new interviews for each position	Adding new interviews and checking for them in both the server and the application	Criteria Met

Ability to change vacancy of an interview if a new candidate wishes to join a vacant interview	Adding candidates to vacant interviews and checking if this reflects in the server and home page	Only partially met because of function
Ability to edit position details	Editing positions and ensuring the changes reflect on the home page and server	Only partially met because of UI error
Ability to delete interviews	Deleting interviews and checking for the changes	Criteria Met
Ability to close positions	Closing positions and checking for the changes	Criteria Met
Ability to log out of application	Logging out of application and ensuring there is no way to access data until logged back in	Criteria Met

Code:

App.js

```
//Importing libraries for the application
import React from 'react';
//The following import is for GUI components from the Bootstrap library
import {
  Container,
  Navbar,
  Nav,
  Card,
  Form,
  Button
} from 'react-bootstrap';
//The following import is from the React router which allows us to browse
components
import {
  BrowserRouter as Router,
  Switch,
  Route,
} from "react-router-dom";
//The following import is to initialize the firebase app which connects to our
server
```

```
import { initializeApp } from 'firebase/app';
//The following import is for user authentication
import {getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut} from
"firebase/auth";
//The following import is for writing, reading and updating the database
import{getFirestore, updateDoc,  getDocs, collection, addDoc, deleteDoc} from
"firebase/firestore";
//The following imports are for the sub-components of the application
import Home from "./Components/home";
import Add from "./Components/Add";
import Edit from "./Components/Edit";

const firebaseConfig = {
//Is removed to protect client privacy
}

// Initialize Firebase

const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getFirestore();
class App extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      userID: '', //Contains user ID
      positions : [],//Contains the positions that the user is attempting to
fill
      posRef : [], //Contains the server reference links for the different
positions
      authenticated : false, //A boolean variable that tells if the user is
authenticated or not
      user: '', //contains user name
      pass: '' //contains account password
    };
    onAuthStateChanged(auth, (user) => {
      if(user){
      getDocs(collection(db, user.uid)).then((snapshot) => {
        const positions = [];
        const posref = [];
        snapshot.docs.map((document) => {
          positions.push(document.data());
           posref.push(document);
          });
        this.setState({positions : positions, posRef : posref});
      });
      this.setState({authenticated : true, userID : user.uid});
      }
```

```
    });
    this.logOut = this.logOut.bind(this);
    this.signIn = this.signIn.bind(this);
    this.del = this.del.bind(this);
    this.addInterview = this.addInterview.bind(this);
    this.addPos = this.addPos.bind(this);
    this.updatePos = this.updatePos.bind(this);
    this.addCandidate = this.addCandidate.bind(this);
    this.closePos= this.closePos.bind(this);
  }
  del(key, id){
    let tp = this.state.positions;
    tp[key].interviews.splice(id,1);
    const doc = this.state.posRef[key];
    const interviews = tp[key].interviews;
    updateDoc(doc.ref, {
      interviews : interviews,
    })
    this.setState({positions : tp})

  }
  updatePos(key, pos, interviews, clientMail, clientName){
    const doc = this.state.posRef[key];
    updateDoc(doc.ref, {
      position : pos,
      interviews : interviews,
      clientName : clientName,
      clientMail : clientMail
    })
    const tp = this.state.positions;
    tp[key] = {
      position : pos,
      interviews: interviews,
      clientName : clientName,
      clientMail : clientMail
    };
    this.setState({positions : tp})
  }
  signIn(){
    signInWithEmailAndPassword(auth, this.state.user,
this.state.pass).catch((error) => window.alert(error.message));
  }
  logOut(){
    this.setState({authenticated : false});
    signOut(auth);
  }
  addInterview(key, time, status, date, name, mail) {
    let tp = this.state.positions;
```

```
    tp[key].interviews.push({
      time : time,
      status : status,
      Date : date,
      candidateName : name,
      candidateMail : mail
    });
    const interviews = tp[key].interviews;
    const doc = this.state.posRef[key];
    updateDoc(doc.ref , {
      interviews : interviews
    });
    this.setState({positions : tp});
  }
  closePos(key){
    let tp = this.state.positions, tpr = this.state.posRef;
    const doc = this.state.posRef[key];
    tp.splice(key,1); tpr.splice(key,1);
    this.setState({positions : tp, posRef: tpr});
    deleteDoc(doc.ref);
  }
  addCandidate(key, id, candidateName, candidateMail){
    const tp = this.state.positions;
    const interviews = tp[key].interviews;
    interviews[id].candidateMail = candidateMail;
    interviews[id].candidateName = candidateName;
    interviews[id].status = false;
    tp[key].interviews = interviews;
    this.setState({positions : tp});
    const doc = this.state.posRef[key];
    updateDoc(doc.ref , {
      interviews : interviews
    });
  }
  addPos(pos){
    const docref = addDoc(collection(db, this.state.userID), {
      position : pos.position,
      clientName : pos.clientName,
      clientMail : pos.clientMail,
      interviews : pos.interviews,
    });
    this.setState({
      positions: this.state.positions.concat([pos]),
      posRef : this.state.posRef.concat([docref])
    });
  }
  render(){
  return(
```

```
    <div className="App">
      <Navbar bg="primary" variant="dark">
        <Container>
          <Navbar.Brand>Interview Scheduler</Navbar.Brand>
          <Nav className="me-auto">
            <Nav.Link href="/">Home</Nav.Link>
            <Nav.Link href="/add"> Add </Nav.Link>
            <Nav.Link href="/edit"> Edit </Nav.Link>
          </Nav>
          <Nav>
            <Nav.Link onClick={this.logOut}> Sign out </Nav.Link>
          </Nav>
        </Container>
      </Navbar>
      <Container className="align-items-center" style={{display: 'flex',
justifyContent:'center'}}>
        {
          this.state.authenticated?
            <>
              <Router>
                <Switch>
                  <Route exact path="/">
                    <Home
                    closePos={this.closePos}
                    addInterview={this.addInterview}
                    positions={this.state.positions}
                    dele={this.del}
                    addCandidate={this.addCandidate}></Home>
                  </Route>
                  <Route exact path="/add">
                    <Add addPos={this.addPos}></Add>
                  </Route>
                  <Route exact path="/edit">
                    <Edit
                    positions={this.state.positions}
                    updatePos={this.updatePos}></Edit>
                  </Route>
                </Switch>
              </Router>
            </>
          :
          <>
            <Container className="w-100 p-5" style={{maxWidth : "400px"}}>
             <Card className="p-2 ">
              <Card.Title>Log in</Card.Title>
              <Card.Body>
                <Form>
                  <Form.Group>
```

```
                    <Form.Label>Email Adress</Form.Label>
                    <Form.Control
                    type="email"
                    placeholder="Enter Email"
                    onChange={(e) => this.setState({user : e.target.value})}
value={this.state.user}/>
                  </Form.Group>
                  <Form.Group >
                    <Form.Label>Password</Form.Label>
                    <Form.Control
                    type="password"
                    placeholder="Enter Password"
                    onChange={(e) => this.setState({pass : e.target.value})}
value={this.state.pass}/>
                  </Form.Group>
                  <Button variant="primary" className="m-2"
onClick={this.signIn}>
                    Submit
                  </Button>
                </Form>
              </Card.Body>
            </Card>
            </Container>
          </>
        }
      </Container>
    </div>
  );

  }
}

export default App;
```

Home.js

```
import {Card, Button, Container, Row, Col, Modal, Form} from 'react-
bootstrap';
import React from 'react';
class Home extends React.Component {
    constructor(props){
        super(props);
        this.state={
            ads : false,
            time : '',
            status : true,
            key : -1,
            Date : '',
```

```
            candidateMail: '',
            candidateName:'',
        }
    }

    add(){
        this.props.addInterview(
                this.state.key, this.state.time,
                this.state.status, this.state.Date,
                this.state.candidateName, this.state.candidateMail);
        this.setState({
            ads : false,
            time : '',
            Date : '',
            status : true,
            key : -1,
            id: -1,
            candidateMail: '',
            candidateName:'',
        })
    }
    render(){
    console.log(this.props.positions);
    const posList =  this.props.positions.map((position, key) =>{
        let intcard = position.interviews.map((interview, id) =>{
            return(
                <Col key={id}>
            <Card  bg={interview.status? 'info' : 'dark'} text="light"
classNameName="p-2 m-3" >
                <Card.Title>{interview.time}</Card.Title>
                <Card.Body>
                    Date: {interview.Date}
                    <br/>
                    Filled : {interview.status ? 'Vacant' : 'Occupied'}
                    <br/>
                    {!interview.status ?
                    <>
                    Name: {interview.candidateName}
                    <br/>
                    Mail: {interview.candidateMail}
                    </> :
                <Button variant="primary" classNameName='m-2'
                onClick={() => this.setState({adc : true, key : key, id: id})}
>
                    Add Candidate</Button>}
                </Card.Body>
            </Card>
                </Col>
```

```
            );
        });
        return(
        <div key={key}>
            <Row className="m-3">
                <Col>
                    <h2>{position.position}</h2>
                    <h3>{position.clientName}, {position.clientMail}</h3>
                </Col>
                <Col>
                    <Button style={{maxWidth:"200px", alignSelf: "left"}}
                    onClick={() => this.setState({ads : true, key : key})}>
                        + Add Interview
                    </Button>
                    <br/>
                    <Button variant="danger"
                     class="m-3"
                     style={{maxWidth:"200px", alignSelf: "left"}}
                     onClick={() => this.props.closePos(key)}>
                          Close position
                    </Button>
                </Col>
            </Row>
            <Row lg={3}>
                    {intcard}
            </Row>
            </div>
        );
    });
     return(
        <Container>
            <Modal show={this.state.ads}>
                <Modal.Header>
                    <Modal.Title>Add interiew</Modal.Title>
                </Modal.Header>
                <Modal.Body>
                    <Form.Label>Time</Form.Label>
                    <Form.Control type="text"
                    placeholder="Enter Interview Time"
                    value={this.state.time}
                    onChange={(e) => this.setState({time : e.target.value})}/>
                    <Form.Label>Date </Form.Label>
                    <Form.Control
                    type="text"
                    placeholder="Enter Date"
                    value={this.state.Date}
                    onChange={(e) => this.setState({Date :
e.target.value})}></Form.Control>
```

```
                    <div> <Form.Label>Status:</Form.Label> </div>
                    <Form.Check
                        inline
                        label="Occupied"
                        type="radio"
                        name="status"
                        onClick={() => this.setState({status : false})}
                    />
                    <Form.Check
                        inline
                        label="Vacant"
                        type="radio"
                        name="status"
                        onClick={() => this.setState({status : true})}
                    />
                    <br/>
                    {
                        !this.state.status ?
                        <>
                            <Form.Label>Name </Form.Label>
                            <Form.Control type="text"
                            placeholder="Enter Name"
                            value={this.state.candidateName}
                            onChange={(e) => this.setState({candidateName:
e.target.value})}>
                            </Form.Control>
                            <Form.Label>Mail ID</Form.Label>
                            <Form.Control type="text"
                            placeholder="Enter Mail ID"
                            value={this.state.candidateMail}
                            onChange={(e) => this.setState({candidateMail:
e.target.value})}>
                            </Form.Control>
                        </> : <></>
                    }
                </Modal.Body>
                <Modal.Footer>
                    <Button onClick={() => this.setState({ads:false})}
variant="secondary"> Close </Button>
                    <Button onClick={() => this.add()}>Add Interview</Button>
                </Modal.Footer>
            </Modal>
            <Modal show={this.state.adc}>
                <Modal.Header>
                    <Modal.Title>Add Candidate</Modal.Title>
                </Modal.Header>
                <Modal.Body>
                            <Form.Label>Name </Form.Label>
```

```
                            <Form.Control type="text"
                            placeholder="Enter Name"
                            value={this.state.candidateName}
                            onChange={(e) => this.setState({candidateName:
e.target.value})}>
                            </Form.Control>
                            <Form.Label>Mail ID</Form.Label>
                            <Form.Control type="text"
                            placeholder="Enter Mail ID"
                            value={this.state.candidateMail}
                            onChange={(e) => this.setState({candidateMail:
e.target.value})}>
                            </Form.Control>
                </Modal.Body>
                <Modal.Footer>
                    <Button
                    onClick={() => this.setState({adc:false})}
                    variant="secondary">
                    Close
                    </Button>
                    <Button
                    onClick={() => {
                    this.props.addCandidate(this.state.key, this.state.id,
                    this.state.candidateName, this.state.candidateMail);
                    this.setState({adc : false});}}>
                        Add Candidate
                    </Button>
                </Modal.Footer>
            </Modal>
            {this.props.positions.length == 0 ?
            <h3> There are no existing positions. Click on the add tab to add
new positions </h3>
            :
            <>
             {posList}
            </>}
        </Container>
      );
   }
}
export default Home;
```

Add.js

```
import React from 'react';
import {Container, Card, Button,  Col, Form, FormControl, FormGroup,
FormLabel, Row} from 'react-bootstrap';

class Add extends React.Component{
    constructor(props){
        super(props);
        this.state ={
            position : '', //stores the name of the job
            clientMail: '', //stores the mail ID of the client
            clientName: '' //stores the client name
        }
    }
    render(){
        return(
            <Container>
                <Col>
                <Card className="p-2" >
                <Form>
                    <FormGroup className="m-3">
                        <FormLabel>Position</FormLabel>
                        <FormControl value={this.state.position}
                        onChange={(e) => this.setState({position :
e.target.value})}
                         placeholder="Enter Position Name"/>
                        <FormLabel>Client Name</FormLabel>
                        <FormControl value={this.state.clientName}
                        onChange={(e) => this.setState({clientName :
e.target.value})}
                        placeholder="Enter Position Name"/>
                        <FormLabel>Client Mail</FormLabel>
                        <FormControl value={this.state.clientMail}
                        placeholder="Enter Position Name"
                        onChange={(e) => this.setState({clientMail:
e.target.value})} />
                    </FormGroup>
                    <FormGroup>
                           <Button variant="primary"
                           className="m-2"
                           onClick={() => {
                            this.props.addPos({
                                position: this.state.position,
                                clientName : this.state.clientName,
                                clientMail : this.state.clientMail,
                                interviews : []});
                            this.setState({
                                position:'',
                                clientName:'',
```

```
                                clientMail:''})}}>
                                    Add position
                            </Button>
                        </FormGroup>
                    </Form>
                    </Card>
                    </Col>
                </Container>
            );
        }
}

export default Add;
```

Edit.js

```
import React from 'react';
import { Dropdown, Row, Col, Card, Form, Button, FormGroup, FormLabel,
FormControl, Container} from 'react-bootstrap';

class Edit extends React.Component{
    constructor(props){
        super(props);
        this.state={
            ukey : null, //Index of position in positions array
            position: 'Select Position',
            interviews : [],
            status : false,
            time : '',
            clientName : '',
            clientMail : ''
        };
    }
    add(){
        const int = this.state.interviews;
        int.push({status: this.state.status, time : this.state.time});
        this.setState({interviews : int})
    }
    del(key){
        const ti = this.state.interviews;
        ti.splice(key,1);
        this.setState({interviews : ti});
    }
    render(){
        let intcard = this.state.interviews.map((interview, id) =>{
            return(
                <Col key={id}>
```

```
            <Card  bg={interview.status? 'info' : 'dark'} text="light"
className="p-2 m-3" >
                <Card.Title>{interview.time}</Card.Title>
                <Card.Body>Filled : {interview.status ? 'Vacant' :
'Occupied'}</Card.Body>
                <Button variant="danger"  style={{maxWidth : "100px"}}
onClick={()=> this.del(id)}>Delete</Button>
            </Card>
                </Col>
            );
       })
        return(
            <Container>

                 <Row lg={1}>
                <Col>
                <Card className="p-2" >
                <Form>
                    <FormGroup className="m-3">
                        <FormLabel>
                <Dropdown>
                    <Dropdown.Toggle>{this.state.position}</Dropdown.Toggle>
                    <Dropdown.Menu>
                        {
                            this.props.positions.map((data, key) =>{
                                return(
                                    <Dropdown.Item id={key}
                                    onClick={() => this.setState({
                                        position :
this.props.positions[key].position,
                                        interviews :
this.props.positions[key].interviews,
                                        ukey : key,
                                        clientName :
this.props.positions[key].clientName,
                                        clientMail :
this.props.positions[key].clientMail
                                    })}>
                                        {data.position}
                                    </Dropdown.Item>
                                );
                            })
                        }
                    </Dropdown.Menu>
                </Dropdown>
                        </FormLabel>
```

```
                        <FormControl placeholder="Enter Position Name"
value={this.state.position} onChange={(e) => this.setState({position :
e.target.value})} />
                    </FormGroup>
                    <FormLabel>Name</FormLabel>
                    <FormControl placeholder="Enter Client Name"
value={this.state.clientName} onChange={(e) => this.setState({clientName :
e.target.value})}/>
                    <FormLabel>Mail</FormLabel>
                    <FormControl placeholder="Enter Client Mail"
value={this.state.clientMail}  onChange={(e) => this.setState({clientMail:
e.target.value})}/>
                    <FormGroup>
                           <Button variant="primary"
                           className="m-2"
                           onClick={() => this.props.updatePos(
                               this.state.ukey,
                               this.state.position,
                               this.state.interviews,
                               this.state.clientMail,
                               this.state.clientName)}>
                                   Update Position
                            </Button>
                    </FormGroup>
                    <Row>
                        {intcard}
                    </Row>
                </Form>
                </Card>
                </Col>
            </Row>
            </Container>
        );
    }
}

export default Edit;
```