MiniChat Project Cheat Sheet

1. Start / Restart Project Locally

```
cd H:\project\chatapp
npm install
docker start chatapp-mongo
npm run dev
```

Server runs at: http://localhost:3000

MongoDB URI in .env:

```
MONGO_URI=mongodb://localhost:27017/chatapp
SESSION_SECRET=your_secret_here
```

2. Backup / Pause Project

```
docker stop chatapp-mongo
docker stop chatapp-node
```

Backup MongoDB locally:

```
mongodump --uri="mongodb://localhost:27017/chatapp" --out ./backup
```

Stop Azure resources from portal (VM, App Service, DB).

3. Resume Later

```
docker start chatapp-mongo
mongorestore ./backup
npm run dev
```

4. Real-Time Chat Notes

- Socket.io handles live updates.
- Client joins room with user ID.
- Messages emitted to both sender and receiver instantly.
- Past messages loaded via GET /chat/:id.
- No page refresh needed for new messages.

5. Key Files / Folders

- server.js -> main server + Socket.io integration
- routes/auth.js -> Google authentication
- routes/chat.js -> chat page and message fetching
- models/User.js, models/Message.js -> MongoDB schemas
- main/ -> EJS views
- for_all/ -> static files (CSS, JS)
- .env -> environment variables (MONGO_URI, SESSION_SECRET)

6. Tips

- Keep .env secret.
- Commit project to GitHub for versioning.
- Stop services to avoid Azure charges.
- Export MongoDB regularly if pausing for months.