

# Deployment guidance

In this file we document how to deploy the project. Since the app has a frontend and a backend, you'll need to deploy both parts to make it work. We will talk about both in detail.

## 1. Frontend (Android)

1. To achieve functionalities related to map and location search, we use the **Google Maps API**. You can to include your own Google Maps API key in case our API key expires (we will try to make it last longer, so hopefully you don't need to do this step). Navigate to `app/src/debug/res/values/google_maps_api.xml` and replace long string between `<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">` and `</string>` with your API key. Also, since we utilized **Maps SDK for Android** and **Places API** to implement the app, make sure they are both enabled in your API configuration.
2. To support Google Map, please select an emulator that supports **Google Play**, i.e., the ones with a triangle in Google Play column. We recommend **Pixel 4**.
3. Depending on the location you will run the backend, you need to modify the `BASE_URL` field in the `UserSession` class located at `app/src/main/java/com/example/easyteamupfrontend/UserSession.java`. By default, the url is configured to connect to our server where our backend is already running on. **To minimized the efforts you spend, you can just use our backend service and skip everything else in the remainder of the document. But if the process on the server timed out without us noticing it, you will need to go through the set-up process by running the backend on your computer. If there's no reaction after filling all fields in registration page and clicking the sign-up button, it's because the server's malfunction. Please do the following.** Modify the `BASE_URL` field to `http://10.0.2.2:<port>/. Please select an unused port number to substitute in and make sure to use the identical port number later in the following backend set-up.`

## 2. Backend (NodeJS and MySQL)

### NodeJS

1. First make sure you install NodeJS. If not, you can go to <https://nodejs.org/en/download/> to install.
2. Then install the required packages for this project: `express`, `mysql2` and `js-sha256`:

```
npm i express mysql2 js-sha256
```

### MySQL

You also need to set up the MySQL database so that later on data can be written to and retrieved from it.

1. Log into MySQL and do the following:

```
CREATE DATABASE EasyTeamUp;
USE EasyTeamUp;
CREATE TABLE `users` (
  `id` varchar(255) NOT NULL,
  `hashedPassword` varchar(255) NOT NULL,
  `age` int NOT NULL,
  `bio` varchar(255) NOT NULL,
  `image` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
);
CREATE TABLE `events` (
  `name` varchar(255) NOT NULL,
  `description` varchar(255) NOT NULL,
  `ownerId` varchar(255) NOT NULL,
  `eventDate` varchar(255) NOT NULL,
  `duration` int NOT NULL,
  `timeslots` varchar(255) NOT NULL,
```

```

    `dueDate` varchar(255) NOT NULL,
    `publicity` varchar(10) NOT NULL,
    `latitude` double NOT NULL,
    `longitude` double NOT NULL,
    `address` varchar(255) NOT NULL,
    PRIMARY KEY (`name`),
    FOREIGN KEY (`ownerId`) REFERENCES `users`(`id`)
);
CREATE TABLE `participants` (
    `eventName` varchar(255) NOT NULL,
    `userId` varchar(255) NOT NULL,
    `status` int NOT NULL, -- -1: rejected/withdrawn, 0: invited, 1: accepted/registered
    FOREIGN KEY (`eventName`) REFERENCES `events`(`name`),
    FOREIGN KEY (`userId`) REFERENCES `users`(`id`)
);
CREATE TABLE `voteTimeSlots` (
    `eventName` varchar(255) NOT NULL,
    `userId` varchar(255) NOT NULL,
    `startTime` varchar(255) NOT NULL,
    FOREIGN KEY (`eventName`) REFERENCES `events`(`name`),
    FOREIGN KEY (`userId`) REFERENCES `users`(`id`)
);
CREATE TABLE `notifications` (
    `userId` varchar(255) NOT NULL,
    `occasion` varchar(255) NOT NULL,
    `eventName` varchar(255) NOT NULL,
    `info` varchar(255) NOT NULL,
    FOREIGN KEY (`eventName`) REFERENCES `events`(`name`),
    FOREIGN KEY (`userId`) REFERENCES `users`(`id`)
);

```

2. Add your MySQL info to `app.js`. Specifically, at about line 8, you can find this. Please replace the `user` and `password` fields with your own MySQL username and password.

```

const db = mysql.createConnection({
  host: '127.0.0.1',
  port: '3306', // this is typically 3306; check your MySQL port number if not work
  user: 'admin', // replace with your MySQL username
  password: 'easyteamup', // replace with your MySQL password
  database: 'EasyTeamUp'
});

```

3. Go to the very end of `app.js` and change the port number specified at the first parameter in this function to the one you used in the frontend.

```

app.listen(80, () => {
  console.log('Listening on port 80...')
})

```

4. After setting up everything above, we can run the backend using

```
node app.js
```

5. You should see something like this in terminal if things work well:

```

Listening on port 80...
Connected to mySQL...

```