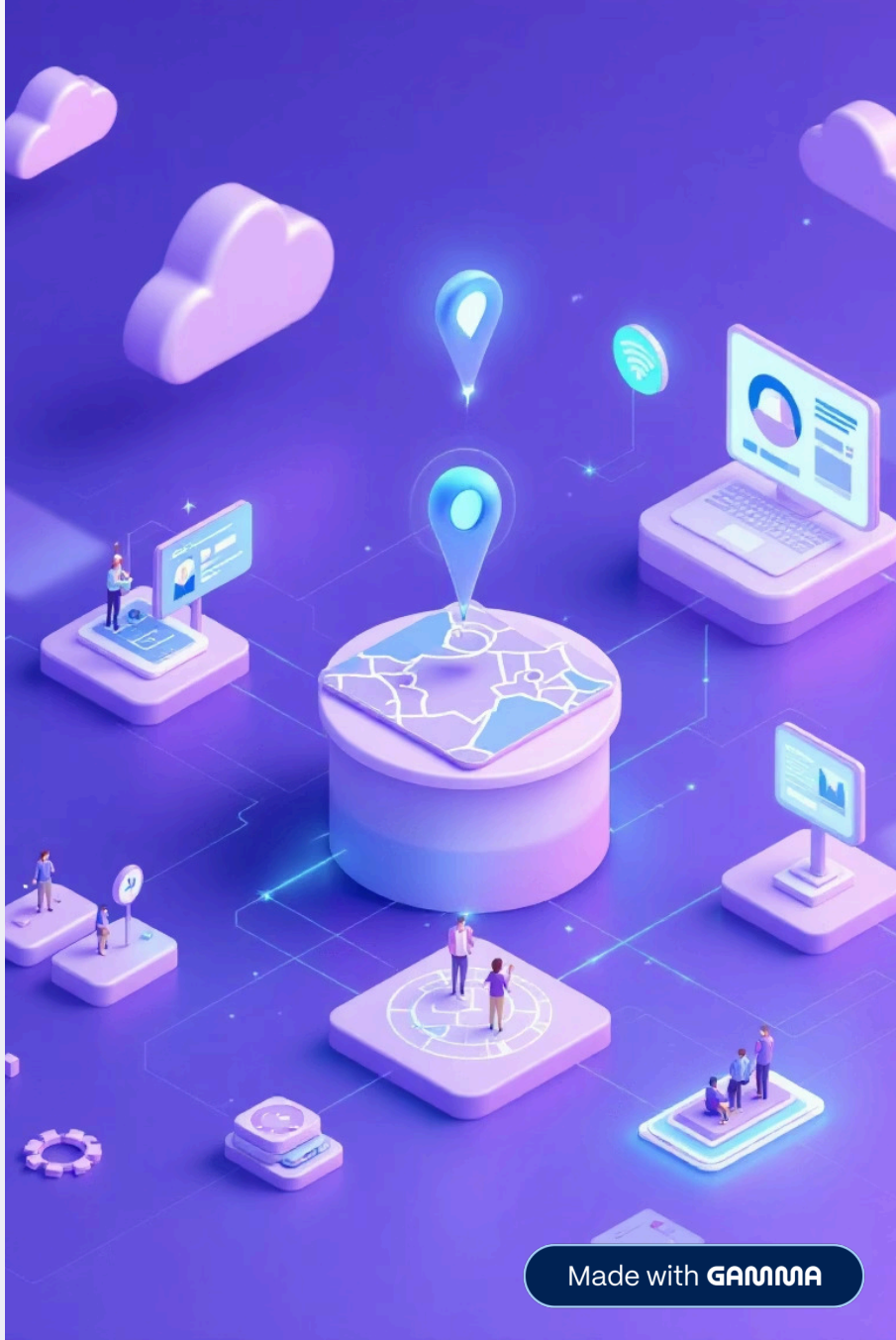
תיעוד מערכת GeoJSON - קשר שרת-לקוח

ברוכים הבאים לתיעוד מערכת ה-GeoJSON שלנו, המתמקדת בקשר בין שרת ללקוח. מערכת זו מאפשרת ניהול יעיל של נתונים גיאוגרפיים, מעקב אחר מיקומים, וחיפוש מבוסס-מיקום - (משרת הנתונים)

במצגת זו נסקור את מבנה הנתונים, נקודות קצה ב-API, שכבות השרת והלקוח. המערכת מיועדת לניהול מידע על קשישים ומתנדבים במרחב גיאוגרפי.

by Eli Pentser E



מבנה נתונים (mongodb)

סכמת מיקום

מבנה נתונים המגדיר נקודה גיאוגרפית עם סוג (Point) וקואורדינטות [קו אורך, קו רוחב] הדורשות ולידציה לערכים תקינים.

אינדקסים גיאוגרפיים

הגדרת אינדקס '2dsphere' על שדה המיקום במודלים של מתנדבים וקשישים לאפשר חיפושים גיאוגרפיים יעילים.

ולידציה

בדיקת תקינות הקואורדינטות: קו אורך בין 180- ל-180 מעלות וקו רוחב בין 90- ל-90 מעלות.

```
# תיעוד מערכת GeoJSON - קשר שרת-לקוח

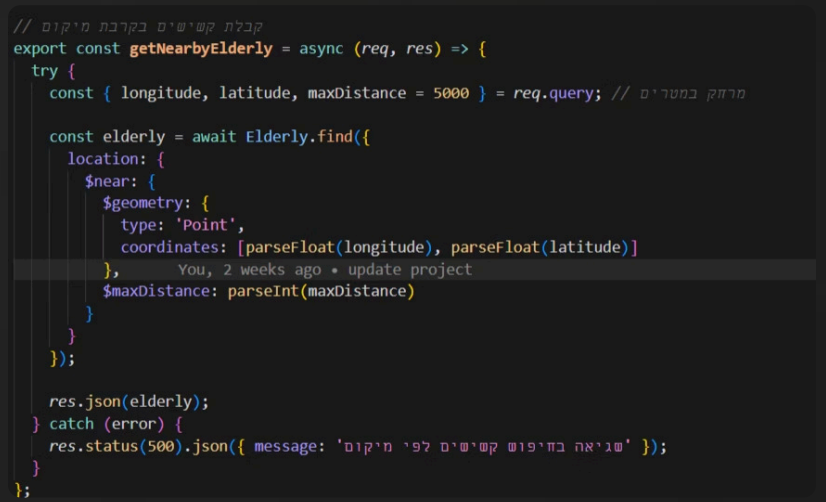
## 1. מבנה נתונים
### 1.1 סכמת מיקום
```javascript
location: {
  type: {
    type: String,
    enum: ['Point'],
    default: 'Point'
  },
  coordinates: {
    type: [Number], // [longitude, latitude]
    required: true,
    validate: {
      validator: function(v) {
        return v.length === 2 &&
               v[0] >= -180 && v[0] <= 180 && // longitu
               v[1] >= -90 && v[1] <= 90;     // latitud
      }
    }
  }
}
```

1.2 אינדקסים גיאוגרפיים
```javascript
// מודל מתנדבים
volunteerSchema.index({ 'location': '2dsphere' });

// מודל קשישים
elderlySchema.index({ 'location': '2dsphere' });
```
```

Made with GAMMA

נקודות קצה ב-API



```
// קבלת קשישים בקרבת מיקום
export const getNearbyElderly = async (req, res) => {
  try {
    const { longitude, latitude, maxDistance = 5000 } = req.query; // מרחק במטרים

    const elderly = await Elderly.find({
      location: {
        $near: {
          $geometry: {
            type: 'Point',
            coordinates: [parseFloat(longitude), parseFloat(latitude)]
          },      You, 2 weeks ago • update project
          $maxDistance: parseInt(maxDistance)
        }
      }
    });

    res.json(elderly);
  } catch (error) {
    res.status(500).json({ message: 'שגיאה בחיפוש קשישים לפי מיקום' });
  }
};
```

קבלת נתוני מפה (למתנדב)

נקודת קצה router.get('/nearby', auth, getNearbyElderly); המקבלת פרמטרים של קו רוחב, קו אורך, רדיוס חיפוש (ברירת מחדל: 5 ק"מ).

עיבוד בקשה

השרת מעבד את הפרמטרים, מבצע חיפוש גיאוגרפי במסד הנתונים, ומחזיר את הנתונים המתאימים לקריטריונים.

החזרת תוצאות

התשובה כוללת רשימות של קשישים והנמצאים בתוך רדיוס החיפוש, עם מידע רלוונטי לכל אחד מהם.

Made with GAMMA

שכבת שרת

חיפוש גיאוגרפי

שימוש ב-$geoWithin ו-$centerSphere למציאת קשישים ומתנדבים בתוך רדיוס מוגדר ממיקום נתון.

המרת כתובת לקואורדינטות

פונקציית geocodeAddress המשתמשת ב-Nominatim של OpenStreetMap להמרת כתובת טקסטואלית לקואורדינטות גיאוגרפיות.

סינון וחישוב

עיבוד נוסף של התוצאות, כולל חישובי מרחק וסינון לפי פרמטרים כמו סטטוס ותאריך ביקור אחרון.

```
### 3.1 חיפוש גיאוגרפי
```javascript
// חיפוש קשישים באזור
const elderly = await Elder.find({
  'location.coordinates': {
    $geoWithin: {
      $centerSphere: [[lng, lat], radiusInDegrees]
    }
  }
});

// חיפוש מתנדבים באזור
const volunteers = await Volunteer.find({
  'location.coordinates': {
    $geoWithin: {
      $centerSphere: [[lng, lat], radiusInDegrees]
    }
  }
});
```

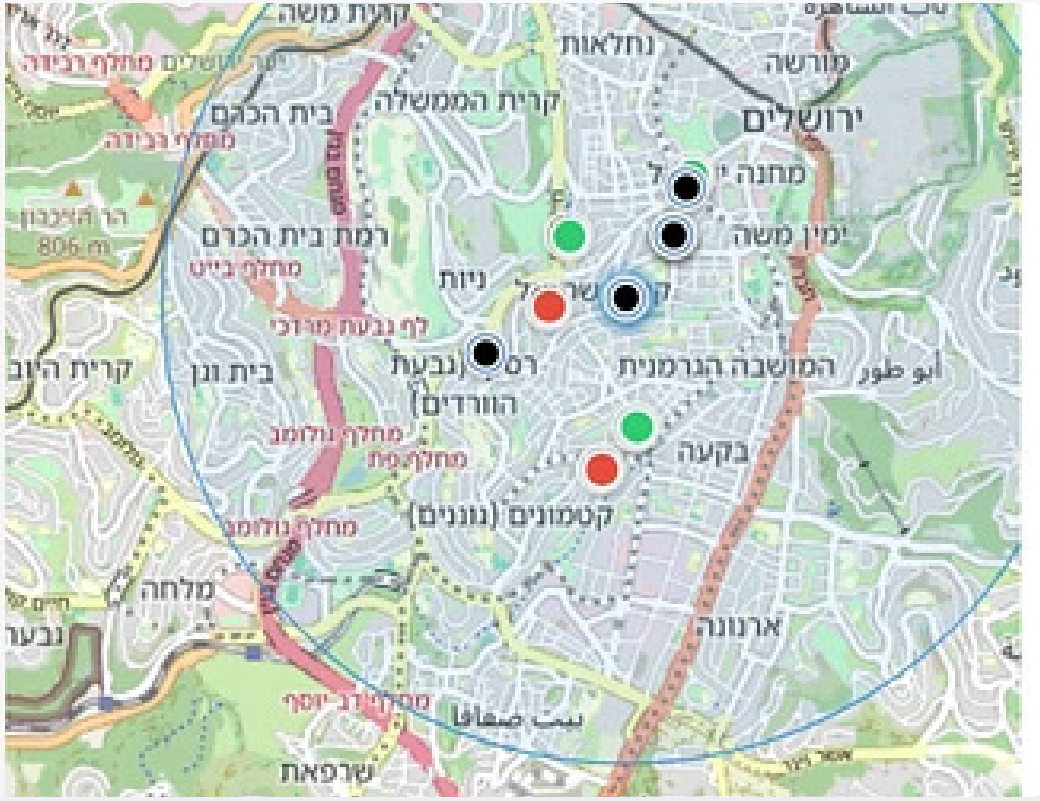
3.2 המרת כתובת לקואורדינטות
```javascript
const geocodeAddress = async (address) => {
  const response = await fetch(
    `https://nominatim.openstreetmap.org/search?format=json&q=${address}, Israel`
  );
  const data = await response.json();
  return data[0] ? [parseFloat(data[0].lon), parseFloat(data[0].lat)] : null;
};
```
```

Made with GAMMA

שכבת לקוח

```
### 4.2 סמנים על המפה
```javascript
// סמן קשיש
<Marker
  position={[elder.location.coordinates[1], elder.location.coordinates[0]]}
  icon={elderlyIcon}
>
  <Popup>
    <div>
      <h3>{elder.firstName} {elder.lastName}</h3>
      <p>סטטוס: {elder.status}</p>
      <p>מרחק: {elder.distanceFromCurrentLocation.toFixed(1)} ק"מ</p>
    </div>
  </Popup>
</Marker>

// סמן מתנדב
<Marker
  position={[volunteer.location.coordinates[1], volunteer.location.coordinates[0]]}
  icon={volunteerIcon}
>
  <Popup>
    <div>
      <h3>{volunteer.firstName} {volunteer.lastName}</h3>
      <p>סטטוס: {volunteer.status}</p>
    </div>
  </Popup>
</Marker>
```



שכבת הלקוח מציגה את המפה באמצעות React Leaflet, עם TileLayer מ-OpenStreetMap. סמנים מותאמים אישית מייצגים קשישים ומתנדבים, כאשר לכל סמן יש חלון קופץ (Popup) המציג מידע רלוונטי כמו שם, סטטוס ומרחק מהמיקום הנוכחי.

המפה מאפשרת אינטראקציה מלאה, כולל הזזה, שינוי מרחק תצוגה (zoom), ולחיצה על סמנים לקבלת מידע נוסף. ניתן גם להציג מעגל המייצג את רדיוס החיפוש הנוכחי.

Made with GAMMA
```

תהליך עבודה מלא

רישום משתמש חדש

קבלת כתובת מהמשתמש, המרתה לקואורדינטות באמצעות Nominatim, ושמירת נקודת GeoJSON במסד הנתונים.

טעינת מפה

קבלת המיקום של המתנדב, שליחת בקשת API עם המיקום והרדיוס, וקבלת רשימות של קשישים באזור.

אדמין (קבלת רשימות של קשישים ומתנדבים לכל אזור)

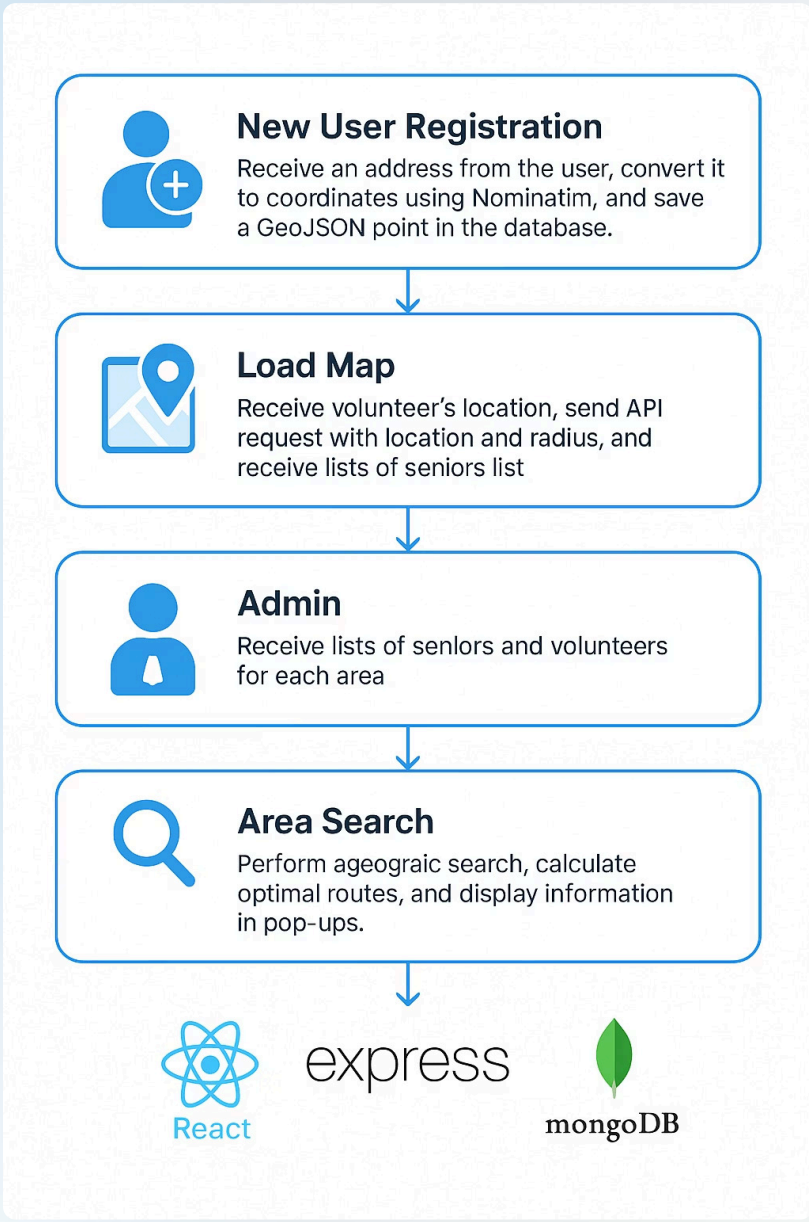
חיפוש באזור

ביצוע חיפוש גיאוגרפי באמצעות $geoWithin ו-$centerSphere, חישוב מרחקים, וסינון לפי פרמטרים נוספים.

צירפתי את הדיאגרמה – היא ממחישה באנגלית את חמשת השלבים, עם אייקונים וכיתוב תמציתי.
אם תרצה שינויים (צבעים, סגנון, כיתובים אחרים או הוספת שלב), אשמח לעדכן!

הצגת תוצאות

הצגת סמנים על המפה, חישוב מסלולים אופטימליים, והצגת מידע בחלונות קופצים.

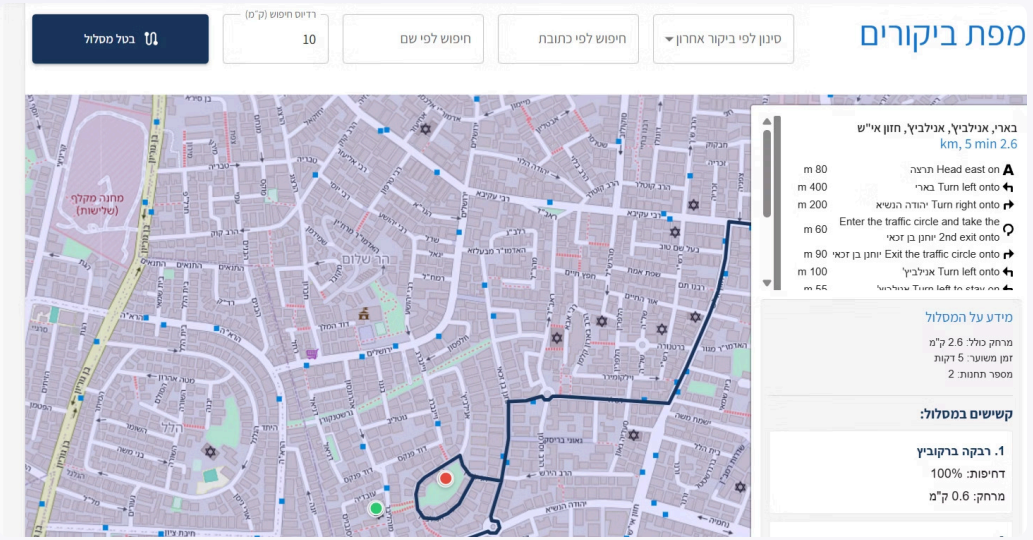


Made with GAMMA

חישוב מסלול אופטימלי

עבור ספריית (plugin) הוא תוסף **Leaflet Routing Machine** המאפשר להוסיף יכולת ניווט, חישוב מסלולים והצגת הוראות **Leaflet**, נסיעה במפות אינטראקטיביות. התוסף תומך במספר נקודות עצירה, מאפשר גרירת נקודות לאורך המסלול, ומציג הוראות נסיעה מפורטות.

התוסף משתמש בשרתים חיצוניים (כגון OSRM או Mapbox) כדי לחשב מסלולים בין נקודות גיאוגרפיות (קואורדינטות).



קוד עבור מסלול אופטימלי עד מתנדבים במסלול

client\src\components\Map\OptimalRoute.jsx

שימוש:

- import 'leaflet-routing-machine';
- import 'leaflet-routing-machine/dist/leaflet-routing-machine.css';

```
  }, [volunteerLocation, elderlyLocations]);

  const calculateOptimalRoute = (elderly, startPoint) => {
    return elderly
      .filter(elder => elder.status === 'פעיל')
      .sort((a, b) => {
        // ... לוגיקת המיון לפי דחיפות ומרחק
      })
      .slice(0, 5); // מקסימום 5 נקודות במסלול
  };

  return (
    <div id="map" style={{ height: '500px', width: '100%' }} />
  );
};

export default OptimalRoute;        You, 2 days ago • update optimal road for valonteer to oldery
```

Made with GAMMA