שאלה 1 (בתחומי הדאטה):

על גבי ה-Dataset IRIS (ניתן לגשת אליו באינטרנט), בצעו את המניפולציות הבאות:

• הוסיפו עמודה שמעגלת את ה-Sepal.length של כל שורה;

• הוסיפו עמודה שערכה בכל שורה יהיה האות הראשונה של המחרוזת בפיצ'ר Iris name אם petal.length גדול מהממוצע של כל ערכי petal.length עבור אותו Iris name; במידה ולא ערך העמודה יהיה האות האחרונה שבמחרוזת הפיצ'ר Iris name.

• הכינו היסטוגרמה של הערך Petal.length.

בונוס:

1. בצעו ניתוח PCA (Principal Component Analysis) על הנתונים והציגו את התוצאות בגרף דו-ממדי.

2. יישמו אלגוריתם clustering (כגון K-means) על הנתונים והשוו את התוצאות לסיווג המקורי של הפרחים.

3. בנו מודל חיזוי (למשל, באמצעות Random Forest) לסיווג סוג הפרח ובדקו את הדיוק שלו באמצעות cross-validation.

שאלה 2 (בתחומי תכנון מערכות):

בשאלה זו נתכנן ארכיטקטורה למערכת אחזור על מיליוני כתבות ותחקירים ממקורות אינטרנטיים.

הרכיבים המרכזיים שמרכיבים את המערכת –

• מנוע אחזור מבוסס חיפוש סמנטי;

• מסד נתונים שמכיל את תוכן הכתבות ופרטי המסגרת;

• רכיבי Frontend ו-Backend.

עליכם להציג דיאגרמה של המערכת עם הרכיבים השונים תוך שאתם מסבירים איך הייתם ממשים זאת ומה השיקולים שהובילו לכך. שימו דגש על ארכיטקטורה גמישה דיו שתאפשר לשנות/לשדרג רכיבים בצורה נוחה ומודולרית.

בונוס:

1. תכננו מערכת ניטור ואיתור תקלות (monitoring and alerting) עבור הארכיטקטורה שהצעתם.

2. הציעו פתרון לגיבוי ושחזור נתונים במקרה של כשל מערכת.

3. תארו כיצד תממשו מנגנון לעדכון אוטומטי של המודל הסמנטי כאשר מתווספים נתונים חדשים.

שאלה 3 (בתחום האלגוריתמיקה):

הסבירו, במילים שלכם, מהו האלגוריתם Binary Search.

בונוס:

1. תארו מצב שבו Binary Search עלול להיכשל או לתת תוצאות לא נכונות. כיצד ניתן להתמודד עם מצב זה?

2. השוו את הביצועים של Binary Search לאלגוריתמים אחרים כמו Linear Search ו-Interpolation Search. באילו מצבים כל אחד מהם יהיה עדיף?

3. הציעו ומתארו גרסה מקבילית (parallel) של אלגוריתם Binary Search. כיצד זה יכול לשפר את הביצועים במערכות מרובות ליבות?

שאלה 4 (בתחום הפרונט-אנד):

פתחו דף HTML עם JavaScript שמציג לוח שנה חודשי. הדף צריך לכלול את הפונקציונליות הבאה:

1. הצגת לוח שנה של החודש הנוכחי.

2. כפתורים למעבר בין חודשים (קדימה ואחורה).

3. סימון היום הנוכחי בצבע שונה.

4. אפשרות להוסיף אירוע ליום מסוים על ידי לחיצה על התא המתאים.

בונוס:

1. הוסיפו אפשרות לשמירת האירועים ב-localStorage כך שיישמרו גם לאחר רענון הדף.

2. יצרו תצוגה שבועית בנוסף לתצוגה החודשית, עם אפשרות מעבר ביניהן.

3. הוסיפו אנימציות CSS למעברים בין חודשים ולהוספת אירועים.

4. יישמו תמיכה בלוח שנה עברי במקביל ללועזי, כולל חגים ומועדים.

5. הוסיפו אפשרות לייצוא האירועים לקובץ iCal או לסנכרון עם Google Calendar.

שאלה 5 (בתחום הבאק-אנד):

פתחו שרת Node.js עם Express שמיישם API לניהול משימות (Todo list). השרת צריך לתמוך בפעולות הבאות:

1. הוספת משימה חדשה (POST /tasks)

2. קבלת רשימת כל המשימות (GET /tasks)

3. עדכון סטטוס משימה (PUT /tasks/:id)

4. מחיקת משימה (DELETE /tasks/:id)

השתמשו ב-MongoDB לאחסון הנתונים.

בונוס:

1. הוסיפו אימות משתמשים (authentication) באמצעות JWT.

2. יישמו מנגנון של rate limiting כדי להגביל את מספר הבקשות מכל IP.

3. הוסיפו אפשרות לתיעוד (logging) של כל הפעולות שמתבצעות במערכת.

4. יישמו מנגנון קשינג (caching) לשיפור ביצועים, למשל באמצעות Redis.

5. הוסיפו תמיכה במשימות חוזרות (recurring tasks) עם אפשרות להגדיר תדירות (יומי, שבועי, חודשי).

6. יישמו מערכת הרשאות (authorization) המאפשרת שיתוף משימות בין משתמשים שונים.