הפרויקט מורכב מ-3 חלקים:

* צד שרת - נכתב בשפת #C.NET .
* התחברות למסד נתונים – Sql Server .
* צד לקוח – נכתב בשפת React .

**צד שרת:**

השרת נכתב בארכיטקטורה שמחלקת את הפרויקט לרכיבים. חלוקת הקוד לרכיבים תוחלט לפי תפקיד הקוד - מה הפעילות שהוא אחראי לבצע.  
באפליקצית Web API סטדנרטית יש לנו שלושה סוגי פעילות: התממשקות מול הקליינט, ניהול הלוגיקה העסקית, ועבודה מול מסד הנתונים.

עבור כל סוג פעילות, יש לנו שכבה נפרדת שאחראית על ביצוע הפעילות.  
כמובן, יש קשר בין השכבות שמעבירות מידע ומפעילות זו את זו.

ארגון הקוד בצורה ארכיטקטונית נותן לנו פיתוח מהיר ויעיל יותר, קוד מובן וקריא, אפשרות לתחזוקה ושדרוג של הקוד בקלות, אבטחה ובטיחות, ועוד...

שם הsolution הוא HMO שזה שם הפרויקט, ובתוכו יש לנו 4 פרויקטים:  
1. HMO.API [WebAPI]- השכבה החיצונית, אחראית על התממשקות מול הקליינט. (נקראת Presentation למרות שאין לה ממשק ויזואלי, אך היא חשופה לקריאות מבחוץ ולכן היא 'תצוגתית')

2. HMO.Core [ClassLibry]- אחראית על המודלים והממשקים של כלל המערכת.

3.HMO.Data [ClassLibry]- אחראית על חיבור למקור הנתונים.

4.HMO.Service [ClassLibry]- אחראית על הלוגיקה העסקית.

פירוט על כל אחד מהשכבות:

HMO.API

פרויקט ה Web API הוא השכבה שחשופה החוצה ומקבלת את בקשות צד הלקוח.

בפרויקט זה יש תיקיית controllers שבתוכה יש מחלקות מסוג Controller שהם בעצם הendPointים שחשופים החוצה וניתן לשלוח אליו קריאה באמצעות ניתוב ייחודי משלו.

בנוסף קיימת תיקיית הModels שבה יש מחלקות עבור האובייקטים אך בצורה של postModel- הצורה שבה אני רוצה שהאובייקט יתקבל כקלט מהמשתמש.

וכן תקיית Mapping עבור המיפויים מהpostModel אל הצורה שבה מוגדר האובייקט בCore.

וכמובן קובץ ה program שזהו הקובץ הראשי של התוכנית הוא מהווה נקודת הכניסה ליישום, כי בו מבצעים את הפקודה לבניית האפליקציה, וכן בו מנהלים את התלויות, תצורת שירות ותצורת מסד נתונים, וכו.....

HMO.Core

שכבת ה Core היא ליבת המערכת. בה נמצאות הגדרות מבני הנתונים והגדרות הפונקציונליות. שכבת ה Core לא מכילה מימוש כלל.  
כל השכבות מכירות את שכבת ה Core והן אלו שמממשות בפועל את הממשקים וההתנהגות שהיא מגדירה.

בשכבה זו קיימים לנו מספר תיקיות:

תיקיית ה Entity: בתיקייה זו יהיו המחלקות שמחזיקות את הנתונים. כל מחלקה ממופת לטבלה אחרת, הEntities הם מחלקות פשוטות שמגדירות את מבני הנתונים בלבד, ולכן יש בהם רק Properties.

תיקיות ה Repository וה :Service בשביל שנוכל להשתמש בהזרקות קוד לצורך גמישות המערכת, אנו יוצרים לכל מחלקה פונקציונלית ממשק, ובמהלך כל הקוד נשתמש בטיפוסי ממשק, בבניית האפליקציה נזריק את הסרברים המתאימים.

ובנוסף תיקיית DTOs שמכילה מחלקות עם האובייקטים רק בצורה שבה נרצה להחזיר את האובייקט בבקשת הGet .

וכן תקיית Mapping שיורשת מProfile ואחראית על מיפוי האובייקט מהEntity אל אובייקט הDTO.

HMO.Data

פרויקט ה data הוא השכבה המקשרת את השרת אל מקור הנתונים,

פרויקט זה מכיל קובץ שנקרא DataContext שהוא יורש מספריית DbContext ,ומכיל בתוכו את מחרוזת ההתחברות למסד הנתונים, וכן מוגדרים בו המערכים מסוג האובייקטים שבCore שאליהם יתקבלו הערכים מהמסד נתונים.

בנוסף פרויקט זה מכיל תקיית Repository שבה יש מחלקות עם פונקציות המתקשרות למסד נתונים. (הם בעצם עובדות מול הDataContext).

HMO.Service

אחראית על הלוגיקה העסקית, בשכבה זו קיימים מחלקות פונקציונאליות שמבצעות את הלוגיקה העסקית בפרוייקט.