API Choices

* Automapper Implementation

Easy data transfer between data classes to dto classes.

Complex logic mapping for future needs.

* Command & Query Pattern

Removing complexity to more managable smaller chunks

Distinct difference between naming, more readable.

Removing DbContext from Controller and, create more flexible structure for different implementations(ex. Dapper ORM for bottlenecks)

* EF – InMemoryDatabase Implementation

Simple implementation and easy to change which database you want to choose in program file with couple of lines.

Multiple most used database support without changing code implementation

* Paycheck calculation without Decorator Pattern

Multiple decorators with only on line of code ( ex. High income cost will be on line. Possible future adaptation if complex logic needed in calculations)

* Multiple test cases for paycheck.
  + Over 80k

An employee with one dependent who is under 50 years old.

An employee with multiple dependents who are all under 50 years old.

An employee with one dependent who is over 50 years old.

An employee with multiple dependents, including one over 50 years old.

An employee with multiple dependents, including multiple dependents over 50 years old.

* + Under 80k (Same with Over80k (– 20k))

An employee with one dependent who is under 50 years old.

An employee with multiple dependents who are all under 50 years old.

An employee with one dependent who is over 50 years old.

An employee with multiple dependents, including one over 50 years old.

An employee with multiple dependents, including multiple dependents over 50

Note: Example calculations in excel file.