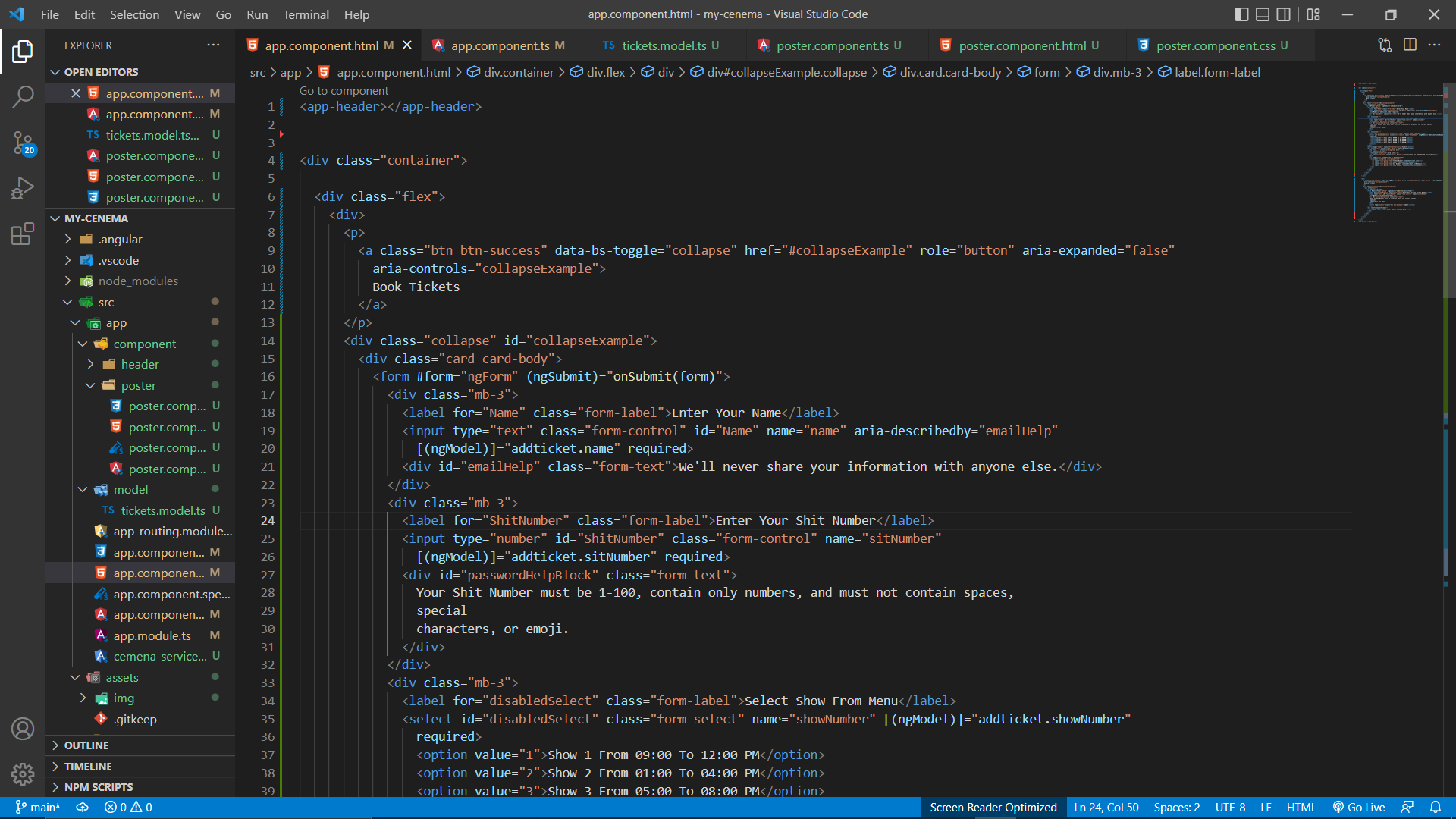
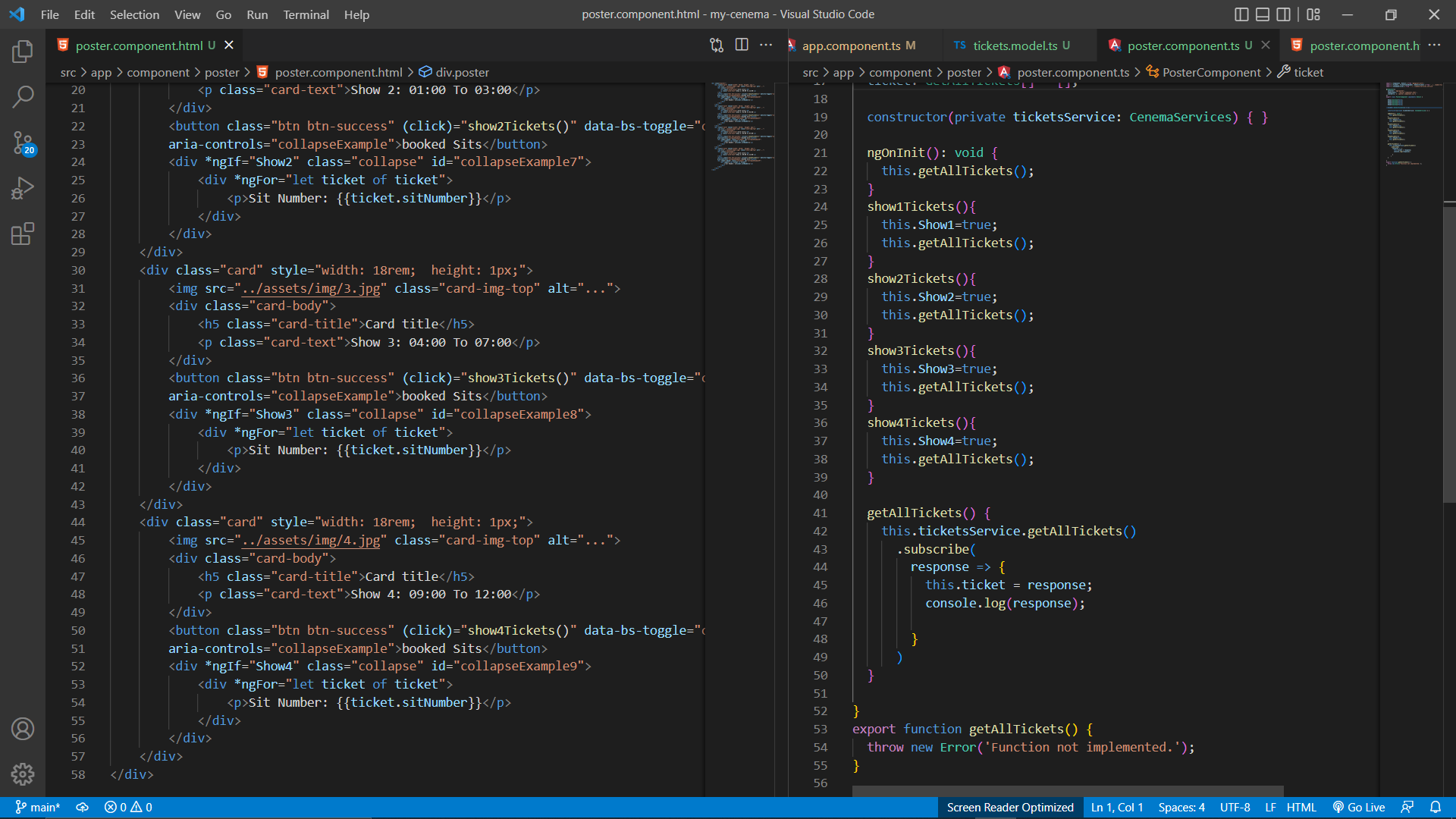
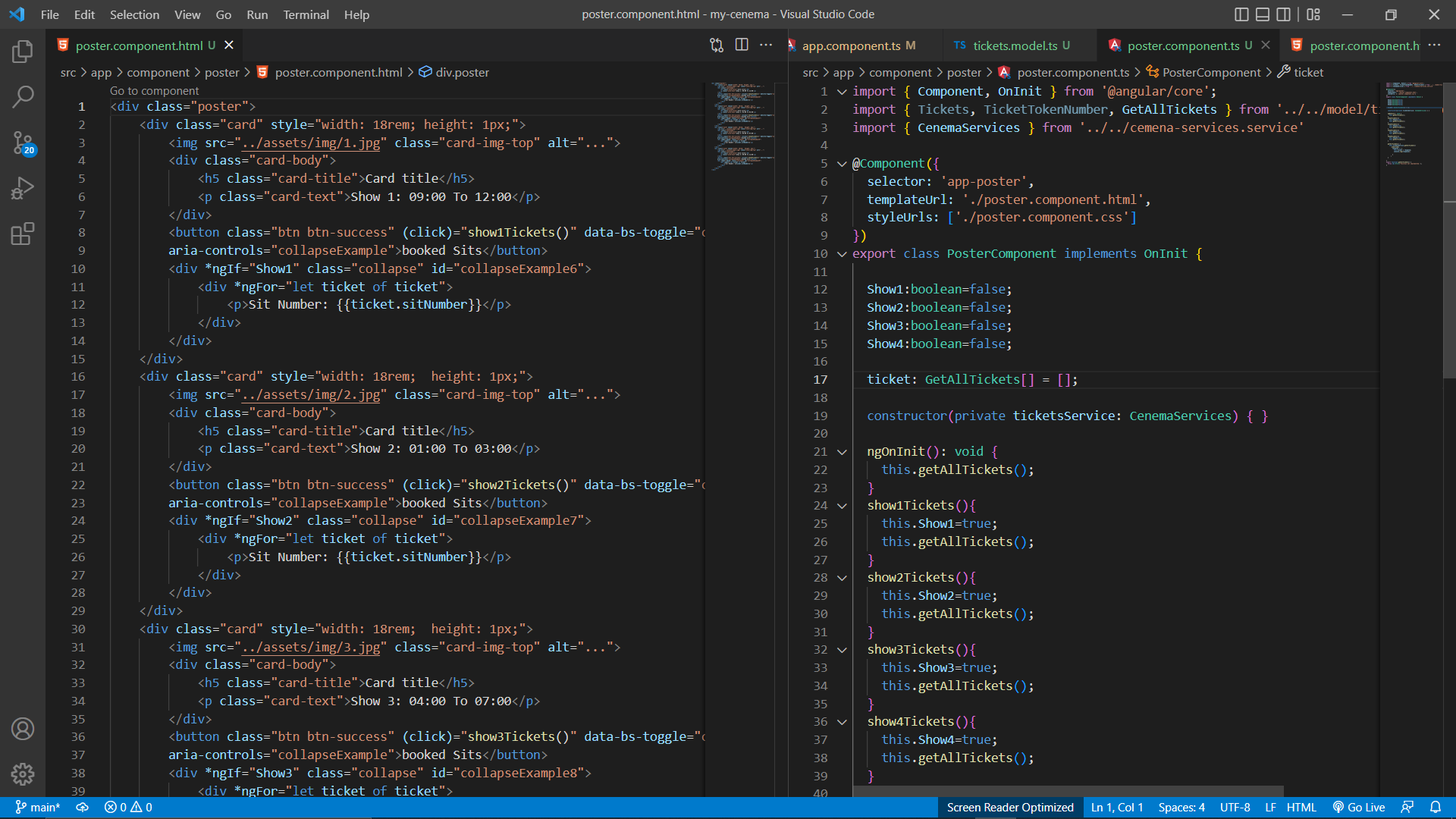
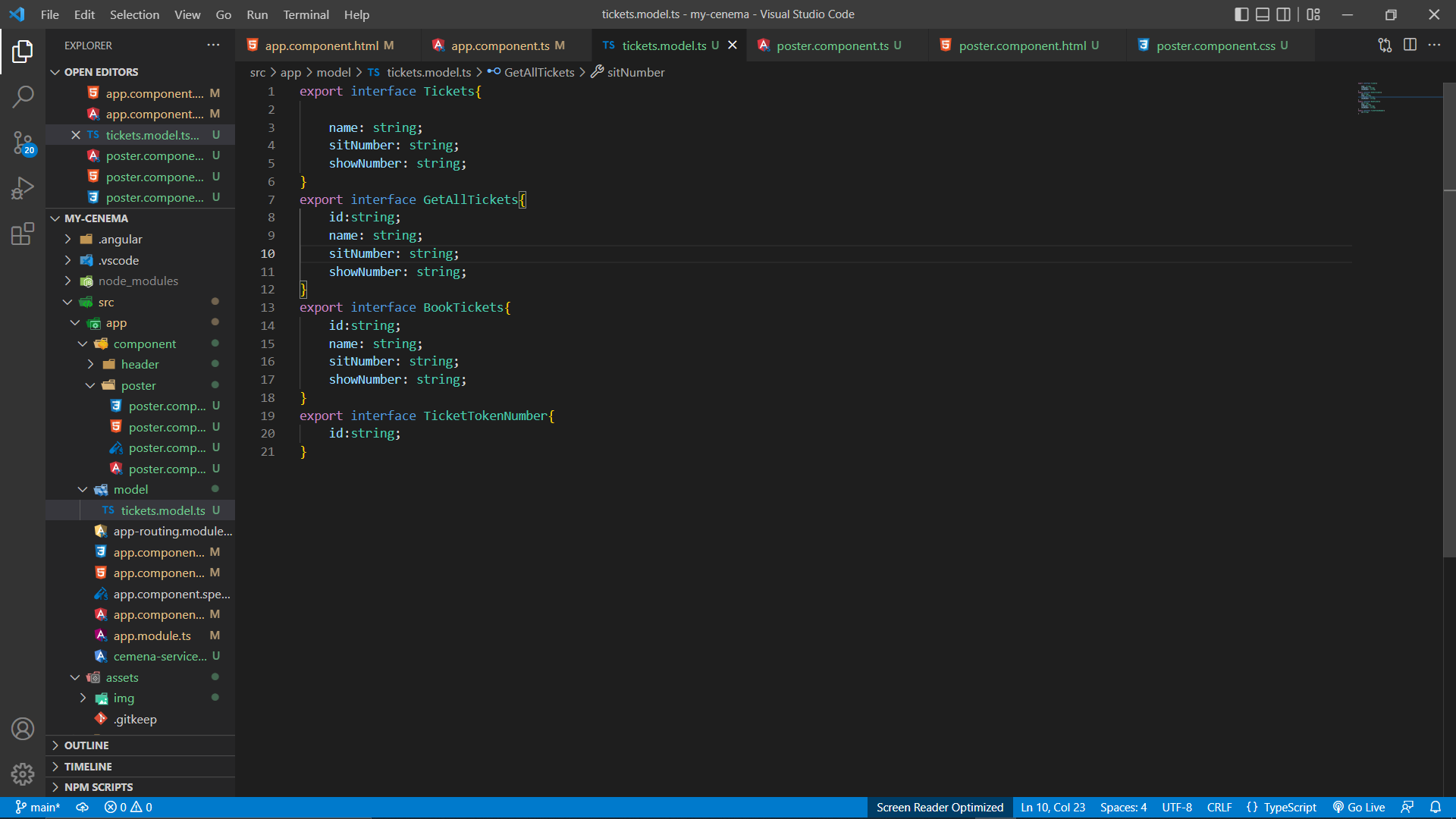
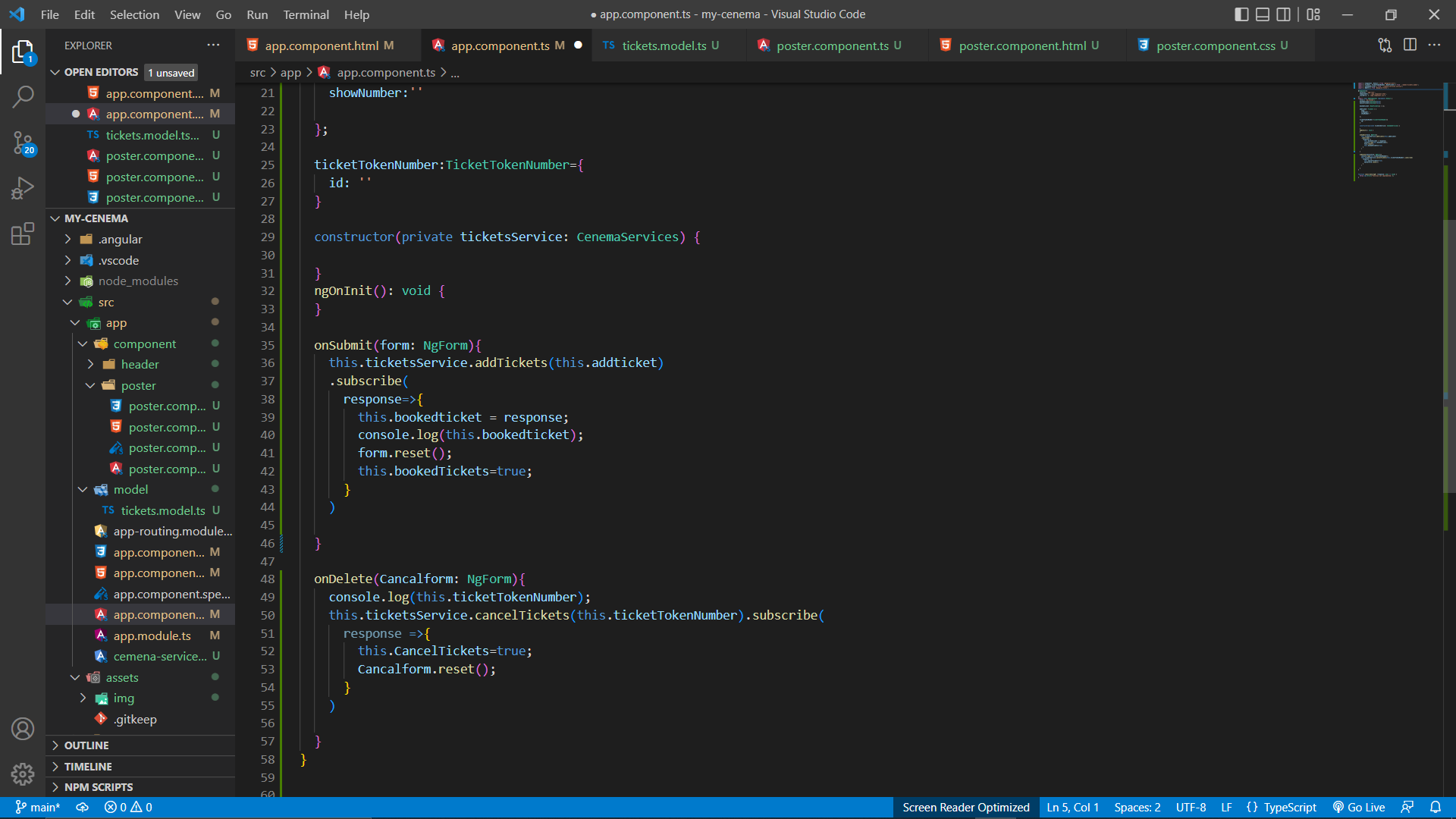
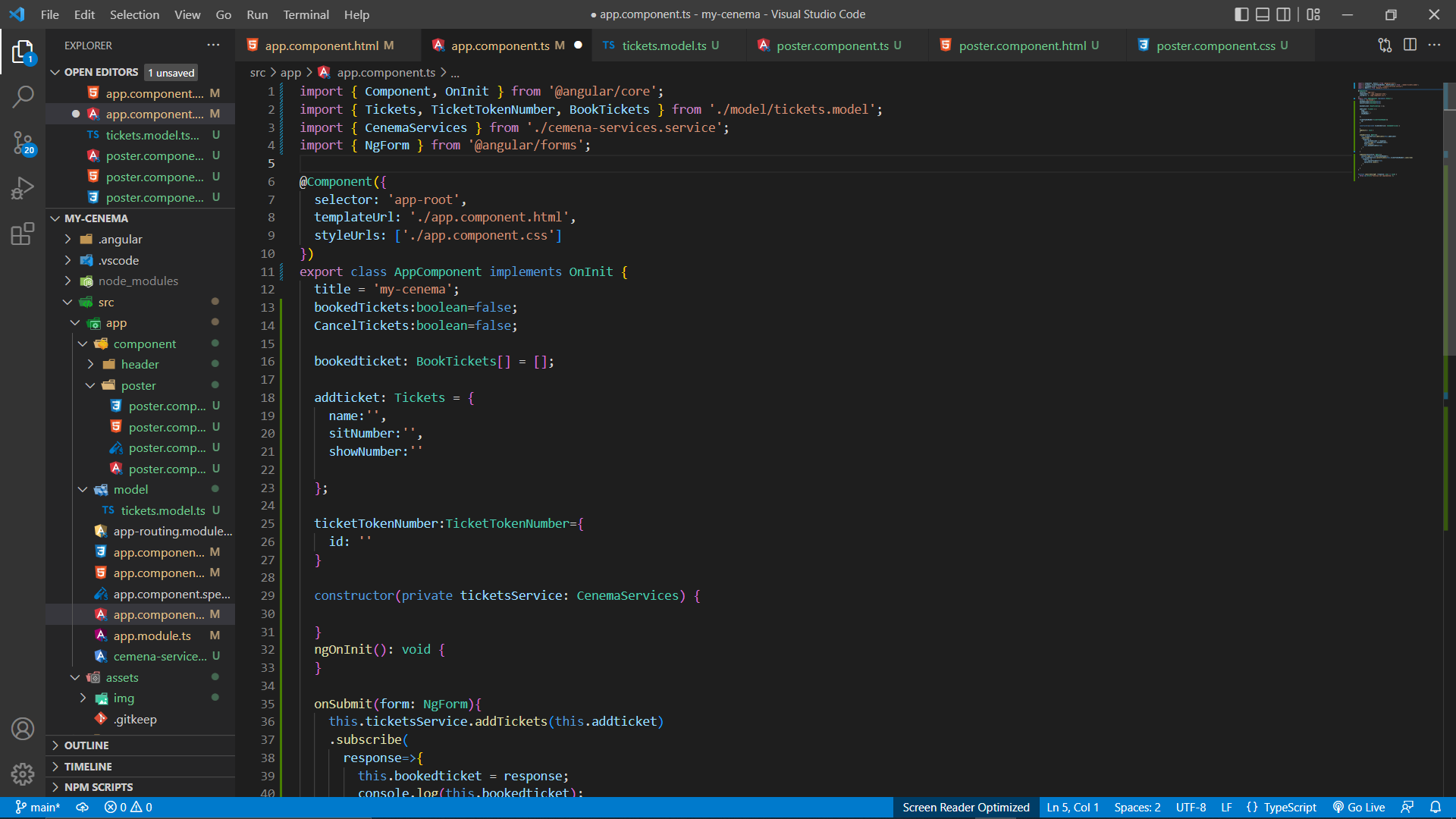
**Ticket Booking System using web API services**

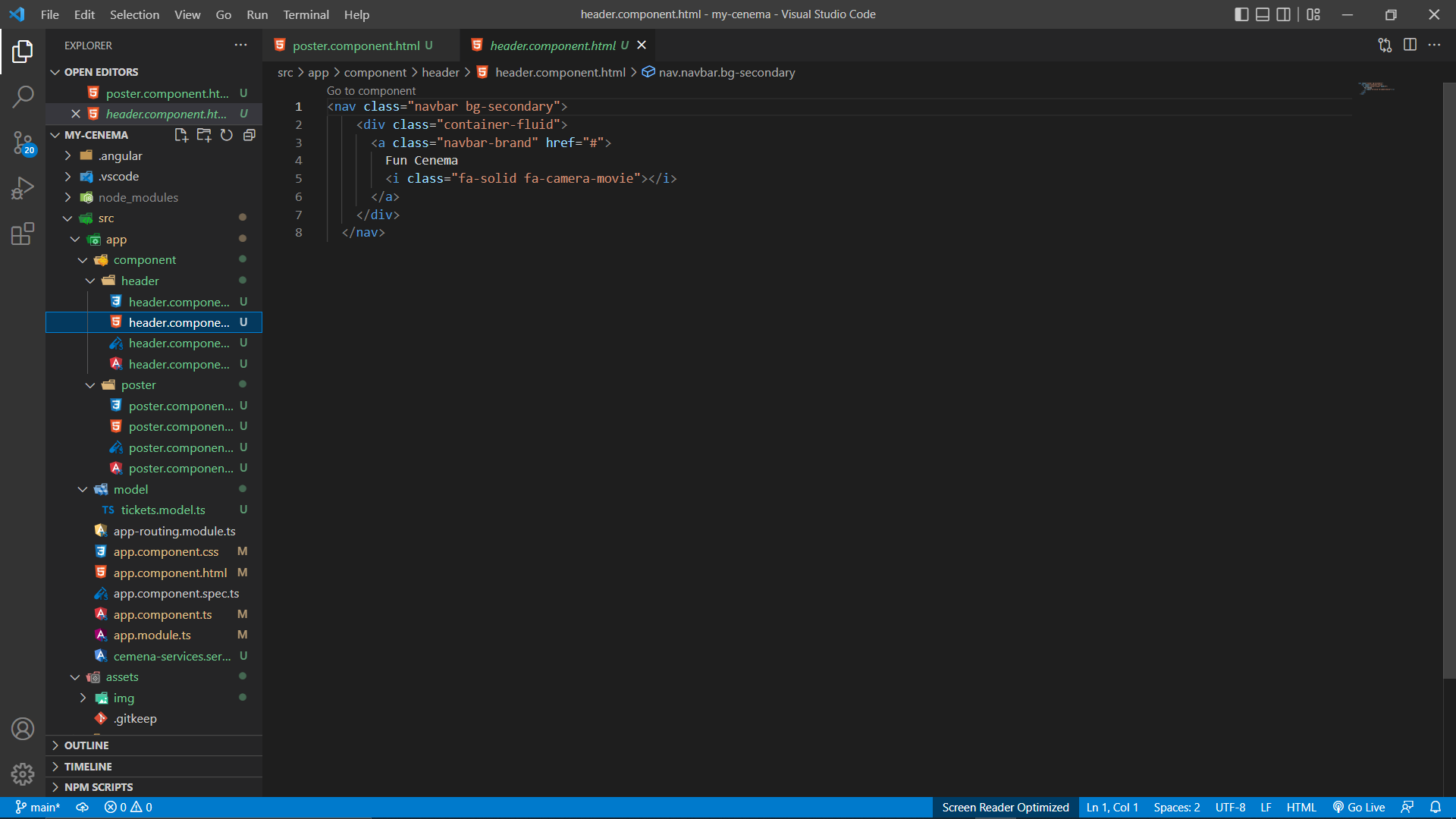
* **Code API services:-**
* using Microsoft.AspNetCore.Mvc;
* using Microsoft.EntityFrameworkCore;
* using System.Linq;
* using TicketsBookingSystem.Data;
* using TicketsBookingSystem.Models;
* namespace TicketsBookingSystem.Controllers
* {
* [ApiController]
* [Route("api/[Controller]")]
* public class TicketsController : Controller
* {
* private TicketsDbContext ticketsDbContext;
* public TicketsController(TicketsDbContext ticketsDbContext)
* {
* this.ticketsDbContext = ticketsDbContext;
* }
* //get All Tickets Details
* [HttpGet]
* public async Task<IActionResult> GetAllTickets()
* {
* var tickets = await ticketsDbContext.Ticketss.ToListAsync();
* return Ok(tickets);
* }
* //get All tickets Show wise
* [HttpGet]
* [Route("{showNumber:int}")]
* //[Route("{id:guid}")]
* [ActionName("GetTAllickets")]
* public async Task<IActionResult> GetTAllickets([FromRoute] int showNumber)
* {
* var ticket = await ticketsDbContext.Ticketss.FirstOrDefaultAsync(x => x.ShowNumber == showNumber);
* if (ticket != null)
* {
* return Ok(ticket);
* }
* return NotFound("Tickets Not Found !");
* }
* //Add Tickets
* [HttpPost]
* public async Task<IActionResult> AddTickets([FromBody] Tickets tickets)
* {
* tickets.Id = Guid.NewGuid();
* await ticketsDbContext.Ticketss.AddAsync(tickets);
* await ticketsDbContext.SaveChangesAsync();
* return Ok(tickets);
* // return CreatedAtAction(nameof(GetTickets), new {id = tickets.Id}, tickets);
* }
* //Updating a Tickets
* [HttpPut]
* [Route("{id:guid}")]
* public async Task<IActionResult> UpdateTicket([FromRoute] Guid id, [FromBody] Tickets tickets)
* {
* var existingticket = await ticketsDbContext.Ticketss.FirstOrDefaultAsync(x => x.Id == id);
* if (existingticket != null)
* {
* existingticket.Name = tickets.Name;
* existingticket.SitNumber = tickets.SitNumber;
* existingticket.ShowNumber = tickets.ShowNumber;
* await ticketsDbContext.SaveChangesAsync();
* return Ok(existingticket);
* }
* return NotFound("Tickets Not Found");
* }
* //Deleteing a Tickets
* [HttpDelete]
* [Route("{id:guid}")]
* public async Task<IActionResult>DeleteTicket([FromRoute] Guid id)
* {
* var existingticket = await ticketsDbContext.Ticketss.FirstOrDefaultAsync(x => x.Id == id);
* if (existingticket != null)
* {
* ticketsDbContext.Ticketss.Remove(existingticket);
* await ticketsDbContext.SaveChangesAsync();
* return Ok(existingticket);
* }
* return NotFound("Tickets Not Found");
* }
* }
* }

Code Angular :-









Output:-

