1. **public** async Task<HttpResponseMessage> PostUserImage()
2. {
3. Dictionary<**string**, **object**> dict = **new** Dictionary<**string**, **object**>();
4. **try**
5. {
7. var httpRequest = HttpContext.Current.Request;
9. **foreach** (**string** file **in** httpRequest.Files)
10. {
11. HttpResponseMessage response = Request.CreateResponse(HttpStatusCode.Created);
13. var postedFile = httpRequest.Files[file];
14. **if** (postedFile != **null** && postedFile.ContentLength > 0)
15. {
17. **int** MaxContentLength = 1024 \* 1024 \* 1; //Size = 1 MB
19. IList<**string**> AllowedFileExtensions = **new** List<**string**> { ".jpg", ".gif", ".png" };
20. var ext = postedFile.FileName.Substring(postedFile.FileName.LastIndexOf('.'));
21. var extension = ext.ToLower();
22. **if** (!AllowedFileExtensions.Contains(extension))
23. {
25. var message = **string**.Format("Please Upload image of type .jpg,.gif,.png.");
27. dict.Add("error", message);
28. **return** Request.CreateResponse(HttpStatusCode.BadRequest, dict);
29. }
30. **else** **if** (postedFile.ContentLength > MaxContentLength)
31. {
33. var message = **string**.Format("Please Upload a file upto 1 mb.");
35. dict.Add("error", message);
36. **return** Request.CreateResponse(HttpStatusCode.BadRequest, dict);
37. }
38. **else**
39. {


43. var filePath = HttpContext.Current.Server.MapPath("~/Userimage/" + postedFile.FileName + extension);
45. postedFile.SaveAs(filePath);
47. }
48. }
50. var message1 = **string**.Format("Image Updated Successfully.");
51. **return** Request.CreateErrorResponse(HttpStatusCode.Created, message1); ;
52. }
53. var res = **string**.Format("Please Upload a image.");
54. dict.Add("error", res);
55. **return** Request.CreateResponse(HttpStatusCode.NotFound, dict);
56. }
57. **catch** (Exception ex)
58. {
59. var res = **string**.Format("some Message");
60. dict.Add("error", res);
61. **return** Request.CreateResponse(HttpStatusCode.NotFound, dict);
62. }
63. }
64. }
65. }