

```
1 module mcq-test-system
2
3 go 1.23.4
4
5 require (
6     github.com/mohae/deepcopy v0.0.0-20170929034955-c48cc78d4826 // indirect
7     github.com/richardlehane/mscfb v1.0.4 // indirect
8     github.com/richardlehane/msoleps v1.0.4 // indirect
9     github.com/xuri/efp v0.0.0-20240408161823-9ad904a10d6d // indirect
10    github.com/xuri/excelize/v2 v2.9.0 // indirect
11    github.com/xuri/nfp v0.0.0-20240318013403-ab9948c2c4a7 // indirect
12    golang.org/x/crypto v0.28.0 // indirect
13    golang.org/x/net v0.30.0 // indirect
14    golang.org/x/text v0.19.0 // indirect
15 )
16
```

```
1 github.com/davecgh/go-spew v1.1.1 h1:vj9j/u1bqnvCEfJ0wUhtl0ARqs3+rkHYY13jYWTU97c=
2 github.com/davecgh/go-spew v1.1.1/go.mod h1:J7Y8YcW2NihsgmVo/mv3lAwL/sk0N4iLHjSsI+c5H38=
3 github.com/mohae/deepcopy v0.0.0-20170929034955-c48cc78d4826 h1:RWengNIwukTxcDr9M+97sNutRR1RKhG9606jWumTTnw=
4 github.com/mohae/deepcopy v0.0.0-20170929034955-c48cc78d4826/go.mod h1:TaXosZuwdSHYgviHp1DAtrULT5eUgsSMsZf+YrPgl8=
5 github.com/pmezard/go-difflib v1.0.0 h1:4DBwDE0NGyQoBHbLQYPwSUPoCMWR5BEzIk/f1lZbAQM=
6 github.com/pmezard/go-difflib v1.0.0/go.mod h1:iKH77koFhYxTK1pcRnkKqfTogsbg7gZNVY4sRDYZ/4=
7 github.com/richardlehane/mscfb v1.0.4 h1:WULscsljNPConisD5hR0+0yZjwK46PfyR6mPu5ZawpM=
8 github.com/richardlehane/mscfb v1.0.4/go.mod h1:YzVpcZg9czvAuhk9T+a3avCpcFPMUWm7gK3DypaEsUk=
9 github.com/richardlehane/msoleps v1.0.1/go.mod h1:BWev5JBpU9Ko2WAgmZEuiz4/u3ZyTKbjLYcmwiWUfWg=
10 github.com/richardlehane/msoleps v1.0.4 h1:WuESlvhX3gH2IHcd8UqyCuFY5yiq/GR/yqaSM/9/g00=
11 github.com/richardlehane/msoleps v1.0.4/go.mod h1:BWev5JBpU9Ko2WAgmZEuiz4/u3ZyTKbjLYcmwiWUfWg=
12 github.com/stretcher/testify v1.8.4 h1:CcVxjf3Q8Pm0mHUKJcdn+eZZtm5yQwehR5yeSVQQcUK=
13 github.com/stretcher/testify v1.8.4/go.mod h1:sz/lmYI0XD/1dqDmKjjqLyZ2RngseejIcXlSw2iwFAo=
14 github.com/xuri/efp v0.0.0-20240408161823-9ad904a10d6d h1:llb0neMWDQe87IzJLS4Ci7psK/lVsJIS2otl+1WyRyY=
15 github.com/xuri/efp v0.0.0-20240408161823-9ad904a10d6d/go.mod h1:ybY/Jr0T0GTCnYjKqmdwxyxn2BQf2RcQIIvex5QldPI=
```

```
16 github.com/xuri/excelize/v2 v2.9.0 h1:1tg0aEq92IOEur1/JfYS/eR0KH0CsRv/rYXXh6YJQE=
17 github.com/xuri/excelize/v2 v2.9.0/go.mod h1:uqey4QBZ9gdMeWApPLdhm9x+
    9o2lq4iVmjiLfBS5hdE=
18 github.com/xuri/nfp v0.0.0-20240318013403-ab9948c2c4a7 h1:
    hPVCafDV85blFTabnqKgNhDckJX25eik94Si9cTER4A=
19 github.com/xuri/nfp v0.0.0-20240318013403-ab9948c2c4a7/go.mod h1:WwHg+CVyzlv/
    TX9xqBFXEZAux0Pxn2k1GNHwG41IIUQ=
20 golang.org/x/crypto v0.28.0 h1:GBDwsMXVQi34v5CCYUm2jkJvu4cbtru2U4TN2PSyQnw=
21 golang.org/x/crypto v0.28.0/go.mod h1:rmgy+3RHxRZMyY0jjAJShp2zgEd0qj2A07U0pYmeQ7U=
22 golang.org/x/image v0.18.0 h1:jGzIakQa/ZXI1IOFxxvaa9W7yP25TqT6cHIHn+6CqvSQ=
23 golang.org/x/image v0.18.0/go.mod h1:4yyo5vMFQjVjUcVk4jEQcU9MGy/ruLF5WvUIlseCM2E=
24 golang.org/x/net v0.30.0 h1:AcW1SDZMkb8IpzCdQUaIq2sP4sZ4zw+55h6ynffypL4=
25 golang.org/x/net v0.30.0/go.mod h1:2wGyMJ5iFasEhkwi13Chk0/t1ECNC4X4eBKKvFYyFfLU=
26 golang.org/x/text v0.19.0 h1:kTxAhCbGbxhK0IwgSKiM05awPoDQ0RpfiVYBfK860YM=
27 golang.org/x/text v0.19.0/go.mod h1:BuEKDfySbSR4drPmRPG/7iBdf8hvFMuRexcpahXiIzY=
28 gopkg.in/yaml.v3 v3.0.1 h1:fxVm/GzAzEWqLHuvctI91KS9hhNmmW0oWu0XTYJS7CA=
29 gopkg.in/yaml.v3 v3.0.1/go.mod h1:K4uyk7z7BCEPqu6E+C64Yfv1cQ7kz7rIZviUmN+EgEM=
30
```

```
1  /* Project Structure
2  ...
3  mcq-test-system/
4  └── cmd/
5      └── main.go
6  └── internal/
7      └── models/
8          ├── question.go
9          ├── report.go
10         ├── handlers/
11             └── test_handler.go
12         └── utils/
13             ├── excel.go
14             └── helpers.go
15     └── templates/
16         ├── start.html
17         ├── question.html
18         └── report.html
19     └── static/
20         ├── css/
21             └── style.css
22     └── go.mod
23     └── go.sum
24     ...
25 */
26
27
```

File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\File_Structure.go

28

29

30

31

32

33

File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\cmd\main.go

```
1 package main
2
3 import (
4     "log"
5     "mcq-test-system/internal/handlers"
6     "net/http"
7     //"path/filepath"
8 )
9
10 func main() {
11     // Initialize handler
12     testHandler := handlers.NewTestHandler()
13
14     // Set up routes
15     mux := http.NewServeMux()
16     mux.HandleFunc("/", testHandler.StartTest)
17     mux.HandleFunc("/question", testHandler.HandleQuestion)
18     mux.HandleFunc("/submit", testHandler.HandleSubmit)
19     mux.HandleFunc("/report", testHandler.GenerateReport)
20
21     // Serve static files
22     fs := http.FileServer(http.Dir("static"))
23     mux.Handle("/static/", http.StripPrefix("/static/", fs))
24
25     // Start server
26     log.Println("Server starting on :8080")
27     log.Fatal(http.ListenAndServe(":8080", mux))
}
```

File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\cmd\main.go

28 }

29

```
1 <!-- static/css/style.css -->
2 .container {
3     max-width: 800px;
4     margin: 0 auto;
5     padding: 20px;
6 }
7
8 .answer-option {
9     margin: 10px 0;
10    padding: 10px;
11    border: 1px solid #ddd;
12    border-radius: 4px;
13 }
14
15 button {
16     padding: 10px 20px;
17     margin: 10px 0;
18     background-color: #007bff;
19     color: white;
20     border: none;
21     border-radius: 4px;
22     cursor: pointer;
23 }
24
25 button:disabled {
26     background-color: #ccc;
27     cursor: not-allowed;
```


File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\internal\utils\excel.go

```
1 // utils/excel.go
2 package utils
3
4 import (
5     "github.com/xuri/excelize/v2"
6     "mcq-test-system/internal/models"
7     "strconv"
8 )
9
10 func LoadQuestionsFromExcel(filename string) ([]models.Question, error) {
11     f, err := excelize.OpenFile(filename)
12     if err != nil {
13         return nil, err
14     }
15     defer f.Close()
16
17     rows, err := f.GetRows("Sheet1")
18     if err != nil {
19         return nil, err
20     }
21
22     var questions []models.Question
23     for i, row := range rows {
24         if i == 0 { // Skip header row
25             continue
26         }
27     }
```

```
28     correctAnswer, _ := strconv.Atoi(row[7])
29     question := models.Question{
30         RawText:      row[0],
31         QuestionText: row[1],
32         Answers:       []string{row[2], row[3], row[4], row[5]},
33         Difficulty:    row[6],
34         CorrectAnswer: correctAnswer,
35         Explanation:   row[8],
36         Explanation2:  row[9],
37         Subject:       row[10],
38         Topic:         row[11],
39     }
40     questions = append(questions, question)
41 }
42 return questions, nil
43 }
44 }
```

File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\internal\utils\helpers.go

1 **package** utils

2

File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\internal\models\report.go

```
1 // models/report.go
2 package models
3
4 import "time"
5
6 type Report struct {
7     NumQuestions int
8     Topic        string
9     CorrectAnswers int
10    ScorePercentage float64
11    TimeTaken       time.Duration
12    AvgTimePerQ     float64
13    StartTime       time.Time
14    EndTime         time.Time
15 }
16
17 type TestSession struct {
18     Questions []Question
19     CurrentIndex int
20     NumQuestions int
21     CorrectCount int
22     StartTime    time.Time
23     EndTime      time.Time
24     Answers      []int // Store student answers
25 }
26
```

File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\internal\models\question.go

```
1 // models/question.go
2 package models
3
4 type Question struct {
5     RawText      string
6     QuestionText string
7     Answers       []string
8     Difficulty    string
9     CorrectAnswer int
10    Explanation    string
11    Explanation2   string
12    Subject        string
13    Topic          string
14 }
15
```

```
1 package handlers
2
3 import (
4     "html/template"
5     "math/rand"
6     "mcq-test-system/internal/models"
7     "mcq-test-system/internal/utills"
8     "net/http"
9     "strconv"
10    "time"
11 )
12
13 type TestHandler struct {
14     templates *template.Template
15     sessions  map[string]*models.TestSession
16 }
17
18 func NewTestHandler() *TestHandler {
19     return &TestHandler{
20         templates: template.Must(template.ParseGlob("templates/*.html")),
21         sessions:  make(map[string]*models.TestSession),
22     }
23 }
24
25 func (h *TestHandler) StartTest(w http.ResponseWriter, r *http.Request) {
26     if r.Method == "GET" {
27         h.templates.ExecuteTemplate(w, "start.html", nil)
```

```
28     return
29 }
30
31 numQuestions, _ := strconv.Atoi(r.FormValue("num_questions"))
32 if numQuestions < 1 {
33     http.Error(w, "Invalid number of questions", http.StatusBadRequest)
34     return
35 }
36
37 questions, err := utils.LoadQuestionsFromExcel("MCQ_question.xlsx")
38 if err != nil {
39     http.Error(w, "Failed to load questions", http.StatusInternalServerError)
40     return
41 }
42
43 // Shuffle questions
44 rand.Shuffle(len(questions), func(i, j int) {
45     questions[i], questions[j] = questions[j], questions[i]
46 })
47
48 // Create new session
49 sessionID := strconv.FormatInt(time.Now().UnixNano(), 10)
50 session := &models.TestSession{
51     Questions:  questions[:numQuestions],
52     StartTime:  time.Now(),
53     NumQuestions: numQuestions,
54     Answers:    make([]int, numQuestions),
```


File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\internal\handlers\test_handler.go

```
55     }
56     h.sessions[sessionID] = session
57
58     http.SetCookie(w, &http.Cookie{
59         Name: "session_id",
60         Value: sessionID,
61         Path: "/",
62     })
63
64     http.Redirect(w, r, "/question", http.StatusSeeOther)
65 }
66
67 func (h *TestHandler) HandleQuestion(w http.ResponseWriter, r *http.Request) {
68     session := h.getSession(r)
69     if session == nil {
70         http.Redirect(w, r, "/", http.StatusSeeOther)
71         return
72     }
73
74     // Check if test time exceeded
75     elapsed := time.Since(session.StartTime)
76     allowedTime := time.Duration(session.NumQuestions) * 2 * time.Minute
77     if elapsed > allowedTime {
78         session.EndTime = time.Now()
79         http.Redirect(w, r, "/report", http.StatusSeeOther)
80         return
81     }
```

```
82
83     if session.CurrentIndex >= session.NumQuestions {
84         session.EndTime = time.Now()
85         http.Redirect(w, r, "/report", http.StatusSeeOther)
86         return
87     }
88
89     currentQ := session.Questions[session.CurrentIndex]
90     // Shuffle answers
91     answers := make([]string, len(currentQ.Answers))
92     copy(answers, currentQ.Answers)
93     rand.Shuffle(len(answers), func(i, j int) {
94         answers[i], answers[j] = answers[j], answers[i]
95     })
96
97     data := struct {
98         Question    models.Question
99         Answers      []string
100        QuestionNum int
101        TimeLeft     int64
102    }{
103        Question:    currentQ,
104        Answers:     answers,
105        QuestionNum: session.CurrentIndex + 1,
106        TimeLeft:    int64(allowedTime-elapsed) / int64(time.Second),
107    }
108
```

```
109 h.templates.ExecuteTemplate(w, "question.html", data)
110 }
111
112 func (h *TestHandler) HandleSubmit(w http.ResponseWriter, r *http.Request) {
113     if r.Method != "POST" {
114         http.Error(w, "Method not allowed", http.StatusMethodNotAllowed)
115         return
116     }
117
118     session := h.getSession(r)
119     if session == nil {
120         http.Error(w, "Session not found", http.StatusBadRequest)
121         return
122     }
123
124     answer, _ := strconv.Atoi(r.FormValue("answer"))
125     correct := answer == session.Questions[session.CurrentIndex].CorrectAnswer
126     if correct {
127         session.CorrectCount++
128     }
129     session.Answers[session.CurrentIndex] = answer
130     session.CurrentIndex++
131
132     w.Write([]byte(`{"correct": ` + strconv.FormatBool(correct) + `}`))
133 }
134
135 func (h *TestHandler) GenerateReport(w http.ResponseWriter, r *http.Request) {
```

```
136 session := h.getSession(r)
137 if session == nil {
138     http.Redirect(w, r, "/", http.StatusSeeOther)
139     return
140 }
141
142 if session.EndTime.IsZero() {
143     session.EndTime = time.Now()
144 }
145
146 timeTaken := session.EndTime.Sub(session.StartTime)
147 avgTimePerQ := float64(timeTaken.Minutes()) / float64(session.NumQuestions)
148 scorePercentage := float64(session.CountCorrect) / float64(session.NumQuestions
149 ) * 100
150
151 report := models.Report{
152     NumQuestions: session.NumQuestions,
153     Topic: session.Questions[0].Topic,
154     CorrectAnswers: session.CountCorrect,
155     ScorePercentage: scorePercentage,
156     TimeTaken: timeTaken,
157     AvgTimePerQ: avgTimePerQ,
158     StartTime: session.StartTime,
159     EndTime: session.EndTime,
160 }
161 h.templates.ExecuteTemplate(w, "report.html", report)
```

File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\internal\handlers\test_handler.go

```
162 }
163
164 func (h *TestHandler) getSession(r *http.Request) *models.TestSession {
165     cookie, err := r.Cookie("session_id")
166     if err != nil {
167         return nil
168     }
169     return h.sessions[cookie.Value]
170 }
171
```

File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\templates\start.html

```
1 <!-- templates/start.html -->
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <title>MCQ Test</title>
6 <link rel="stylesheet" href="/static/css/style.css">
7 </head>
8 <body>
9 <div class="container">
10 <h1>MCQ Test</h1>
11 <form method="POST">
12 <div class="form-group">
13 <label>Number of questions:</label>
14 <input type="number" name="num_questions" required min="1" class="form-control
">
15 <small class="form-text">You will have 2 minutes per question</small>
16 </div>
17 <button type="submit" class="btn-primary">Start Test</button>
18 </form>
19 </div>
20 </body>
21 </html>
```

File - C:\Users\Kaunggo\Go_Projects\mcq-test-system\templates\report.html

```
1 <!-- templates/report.html -->
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <title>Test Report</title>
6 <link rel="stylesheet" href="/static/css/style.css">
7 </head>
8 <body>
9 <div class="container">
10 <div class="report-card">
11 <h1>Test Report</h1>
12
13 <div class="score-highlight">
14     Final Score: {{printf "%.1f" .ScorePercentage}}%
15 </div>
16
17 <div class="stat-row">
18     <span class="stat-label">Topic:</span>
19     <span>{{.Topic}}</span>
20 </div>
21
22 <div class="stat-row">
23     <span class="stat-label">Questions Answered:</span>
24     <span>{{.NumQuestions}}</span>
25 </div>
26
27 <div class="stat-row">
```

```
28 <span class="stat-label">Correct Answers:</span>
29 <span>{{.CorrectAnswers}} out of {{.NumQuestions}}</span>
30 </div>
31
32 <div class="stat-row">
33 <span class="stat-label">Total Time:</span>
34 <span>{{.TimeTaken.Minutes | printf "%.1f"}} minutes</span>
35 </div>
36
37 <div class="stat-row">
38 <span class="stat-label">Average Time per Question:</span>
39 <span>{{printf "%.1f" .AvgTimePerQ}} minutes</span>
40 </div>
41
42 <div class="stat-row">
43 <span class="stat-label">Start Time:</span>
44 <span>{{.StartTime.Format "02/01/2006 15:04"}}</span>
45 </div>
46
47 <div class="stat-row">
48 <span class="stat-label">End Time:</span>
49 <span>{{.EndTime.Format "02/01/2006 15:04"}}</span>
50 </div>
51 </div>
52
53 <div class="button-group">
54 <button onclick="window.location.href='/'" class="btn-primary">Start New Test</
```


File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\templates\report.html

```
54 button>
55 <button onclick="window.print()" class="btn-secondary">Print Report</button>
56 </div>
57 </div>
58 </body>
59 </html>
```

```
1 <!-- templates/question.html -->
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Question {{.QuestionNum}}</title>
6   <link rel="stylesheet" href="/static/css/style.css">
7 </head>
8 <body>
9   <div class="container">
10    <div class="timer">Time Remaining: <span id="timer">{{.TimeLeft}}</span> seconds</div>
11    <h2>Question {{.QuestionNum}}</h2>
12    <p class="question-text">{{.Question.QuestionText}}</p>
13
14    <form id="questionForm">
15      {{range $index, $answer := .Answers}}
16        <div class="answer-option">
17          <input type="radio" name="answer" value="{{ $index }}" id="answer{{ $index }}"
18            required>
19          <label for="answer{{ $index }}">{{ $answer }}</label>
20        </div>
21      {{end}}
22      <button type="submit" id="confirmBtn" class="btn-primary">Confirm</button>
23    </form>
24
25    <div id="feedback" style="display: none;">
26    <p id="resultText"></p>
```

```
26 <p class="explanation">{{.Question.Explanation}}</p>
27 <button id="nextBtn" class="btn-secondary">Next Question</button>
28 </div>
29 </div>
30
31 <script>
32 // Timer functionality
33 let timeLeft = {{.TimeLeft}};
34 const timerElement = document.getElementById('timer');
35 const timerInterval = setInterval(() => {
36   timeLeft--;
37   timerElement.textContent = timeLeft;
38   if (timeLeft <= 0) {
39     clearInterval(timerInterval);
40     window.location.href = '/report';
41   }
42 }, 1000);
43
44 // Form submission
45 document.getElementById('questionForm').onsubmit = function(e) {
46   e.preventDefault();
47   const formData = new FormData(this);
48
49   fetch('/submit', {
50     method: 'POST',
51     body: formData
52   })
```

File - C:\Users\Kaung\go\Go_Projects\mcq-test-system\templates\question.html

```
53     .then(response => response.json())
54     .then(data => {
55         document.getElementById('confirmBtn').disabled = true;
56         document.getElementById('feedback').style.display = 'block';
57         document.getElementById('resultText').textContent =
58             data.correct ? '✓ Correct!' : '✗ Incorrect';
59         document.getElementById('resultText').className =
60             data.correct ? 'correct' : 'incorrect';
61     });
62 };
63
64     document.getElementById('nextBtn').onclick = function() {
65         window.location.href = '/question';
66     };
67 </script>
68 </body>
69 </html>
```