1. Identify and validate high-priority use cases for synthetic monitoring across key portals

Perform an assessment to identify critical application areas where synthetic monitoring would add maximum value — across platforms like Member Portal, Employer Portal, and Claims Portal. Review if existing monitors are sufficient or require enhancement. Gather input on typical user workflows or dummy actions (e.g., login, navigation, or form load) that can simulate real usage. Additionally, identify key performance metrics (such as page load time, response status, and step success) that will help application and SRE teams proactively monitor system health and user experience.

. Configure browser clickpath monitors for payment and billing workflows

Set up synthetic browser clickpath tests to simulate key user payment and billing transactions. This ensures continuous availability checks for critical checkout paths and alerts on functional failures.

---

3. Expand synthetic monitoring to cover top 5 user journeys across business applications

Identify and configure synthetic monitors for the most frequently used and business-critical user journeys. This provides better end-to-end visibility across core application workflows.

---

4. Configure HTTP monitors for backend APIs not currently covered

Enable HTTP synthetic monitors for APIs like /getemployermegamenudetails.json and other backend services that impact UI functionality but are not currently under monitoring.

---

5. Update synthetic monitors failing due to DOM or page structure changes

Review and fix existing synthetic monitors failing due to recent UI/DOM changes. This includes updating element selectors and validating new click paths post-deployment.

---

6. Configure alerting profiles for synthetic test failures with appropriate routing

Define alerting rules and assign them to correct application teams based on the synthetic monitor context. This ensures ownership-based routing and timely action on failures.

---

7. Create dashboards summarizing synthetic test results and status

Build dashboards that visualize real-time and historical status of synthetic monitors. This should include pass/fail trends, frequency, and grouped views by application or service.

---

8. Adjust alert thresholds in synthetic monitors to avoid false positives

Analyze noisy alerts from synthetic monitors and update thresholds (e.g., response time, load time) to reduce false alerts while maintaining meaningful failure detection.

---

9. Configure synthetic monitors in pre-production environments for shakeout validation

Set up synthetic monitors in staging or pre-production to validate application functionality after deployments, ensuring stability before go-live.

---

10. Provide Dynatrace access to application teams for synthetic monitor maintenance

Assign appropriate Dynatrace access roles to application SPOCs, enabling them to manage, edit, and validate synthetic monitors for their services.

---

11. Conduct coverage analysis to identify unmonitored application flows

Compare current synthetic test coverage against the complete application inventory and identify gaps. Create a backlog of missing monitors for future setup.

---

12. Integrate synthetic monitor failures with ServiceNow for automated ticketing

Set up integration between Dynatrace and ServiceNow to automatically create incidents when a synthetic monitor fails, ensuring prompt tracking and response.